



UNIVERSIDAD DE LA RIOJA

TRABAJO FIN DE ESTUDIOS

Título

Representaciones gráficas en Julia de Conjuntos de Julia

Autor/es

IÑAKI ZUAZOLA GARDE

Director/es

LUIS JAVIER HERNÁNDEZ PARICIO y MARÍA TERESA RIVAS RODRÍGUEZ ,

Facultad

Facultad de Ciencia y Tecnología

Titulación

Grado en Matemáticas

Departamento

MATEMÁTICAS Y COMPUTACIÓN

Curso académico

2017-18



Representaciones gráficas en Julia de Conjuntos de Julia, de IÑAKI ZUAZOLA GARDE

(publicada por la Universidad de La Rioja) se difunde bajo una Licencia Creative Commons Reconocimiento-NoComercial-SinObraDerivada 3.0 Unported.

Permisos que vayan más allá de lo cubierto por esta licencia pueden solicitarse a los titulares del copyright.

© El autor, 2018

© Universidad de La Rioja, 2018

publicaciones.unirioja.es

E-mail: publicaciones@unirioja.es



UNIVERSIDAD DE LA RIOJA

Facultad de Ciencia y Tecnología

TRABAJO FIN DE GRADO

Grado en Matemáticas

**Representaciones gráficas
en Julia
de Conjuntos de Julia**

Realizado por:

Iñaki Zuazola Garde

Tutelado por:

María Teresa Rivas Rodríguez

Luis Javier Hernández Paricio

Logroño, junio 2018

Representaciones gráficas en Julia de Conjuntos de Julia



Iñaki Zuazola Garde

Tutores:

Luis Javier Hernández Paricio y María Teresa Rivas Rodríguez

UNIVERSIDAD DE LA RIOJA

Trabajo Fin de Grado en Matemáticas

Agradecer a mis tutores María Teresa Rivas Rodríguez
y Luis Javier Hernández Paricio por todo lo
que me han enseñado y lo que me han ayudado.

A mis padres por su apoyo durante estos años y
especialmente a mis hermanos David, Mario y Alba.

Resumen

El objetivo principal de este trabajo es el estudio de representaciones gráficas de conjuntos de Julia inducidos por la iteración de funciones racionales. Para ello, presentaremos un marco matemático basado en modelos proyectivos que representan a la Esfera de Riemann y a las funciones racionales. También mostraremos algunos algoritmos implementados en el entorno informático Julia para la aproximación gráfica tanto del conjunto de Julia como del de Fatou, que utilizaremos para visualizar algunos ejemplos sugeridos por Milnor asociados a polinomios cuadráticos y cúbicos. Por último, analizaremos el conjunto de Mandelbrot y daremos una representación gráfica obtenida mediante una implementación en el lenguaje de programación Julia.

Abstract

The main goal of this work is the study of graphical representations of Julia sets induced by the iteration of rational functions. For this purpose, we will present a mathematical framework based on projective models representing the Riemann Sphere and rational functions. We will also show some algorithms implemented in the computing environment Julia for the graphical approximation of the Julia and Fatou sets, which we will use to display some examples suggested by Milnor associated with quadratic and cubic polynomials. Finally, we will analyse the Mandelbrot set and we will give a graphical representation obtained from an implementation in Julia language.

Índice General

Resumen	I
Abstract	II
Introducción	1
1. Preliminares	5
1.1. Variedades topológicas	6
1.2. Superficies de clase C^p y analíticas	7
1.3. Ejemplos de Superficies de Riemann	8
1.3.1. La 2-esfera S^2	8
1.3.2. El plano complejo ampliado $\hat{\mathbb{C}}$	9
1.3.3. La recta proyectiva compleja $\mathbf{P}^1(\mathbb{C})$	9
1.3.4. Biyecciones canónicas	10
1.3.5. Métricas cordales	12
1.4. Semiflujos discretos	12
1.4.1. Semiflujos discretos	12
1.4.2. Funciones racionales complejas	13
2. Extensión de modelos y pares homogéneos	16
2.1. Modelos extendidos	16
2.2. Coordenadas homogéneas normalizadas	17
2.3. Aplicaciones racionales complejas y pares homogéneos	18
2.4. Normalización de un par homogéneo	22
3. Algoritmos utilizados para la representación de cuencas de atracción	23
3.1. Algoritmo para el cálculo de puntos fijos	23
3.2. Proceso de iteración de una función racional	28
3.3. Construcción de 2-arrays posición-iteración	29
3.4. Construcción de entornos rectangulares y representaciones gráficas globales	29

4. Ejemplos de Conjuntos de Julia	33
4.1. Funciones racionales hiperbólicas y expansivas	33
4.2. Conjunto de Julia diferencialmente parametrizable	34
4.3. Conjunto de Julia topológicamente parametrizable	35
4.4. Conjunto de Julia de una función racional no expansiva	37
4.5. Conjunto de Julia de tipo Cantoriano	38
4.6. Conjunto de Julia de una función racional expansiva	40
4.7. Conjunto de Julia de un polinomio cúbico	40
4.8. Conjunto de Julia de tipo espiral logarítmica	42
4.9. Conjunto de Julia de tipo arbóreo	44
4.10. Conjunto de Mandelbrot	46
Conclusiones	49
Referencias	50

Introducción

Los objetivos principales de este Trabajo Fin de Grado son:

- i. Descripción de un marco matemático para el desarrollo de algoritmos asociados a la iteración de funciones racionales en la Esfera de Riemann.
- ii. Estudio de representaciones gráficas de las cuencas de atracción correspondientes a una función racional en la Esfera de Riemann.
- iii. Implementación de algoritmos para la representación gráfica de conjuntos de Julia mediante el lenguaje Julia.
- iv. Estudio, mediante aproximaciones gráficas, de los conjuntos de Julia y de Fatou de una función racional.
- v. Estudio gráfico de algunos ejemplos importantes asociados a polinomios cuadráticos y cúbicos presentados en un trabajo de Milnor.
- vi. Elaboración e implementación de algoritmos en Julia para la representación del conjunto de Mandelbrot.
- vii. Implementación en paralelo de algunos de los algoritmos anteriores.

Este trabajo ha sido estructurado en cuatro secciones. A continuación, expondremos brevemente cada una de ellas:

La sección 1 está dedicada a recordar algunos conceptos, estructuras y resultados necesarios para el desarrollo posterior del trabajo. Comenzamos dicha sección con algunos preliminares básicos sobre espacios topológicos generales. Recordamos la noción de variedad topológica, así como algunos conceptos y resultados relacionados con variedades. A continuación, describimos las estructuras que nos permiten establecer un marco matemático de los algoritmos utilizados para la representación de cuencas de atracción de una función racional. Dichas estructuras, comúnmente conocidas como Esfera de Riemann, son: la 2-esfera S^2 , el plano complejo ampliado $\hat{\mathbb{C}}$ y la recta proyectiva compleja $\mathbf{P}^1(\mathbb{C})$. Mostramos, además, las biyecciones canónicas que las relacionan, y usamos dichas biyecciones para describir las métricas que utilizamos en cada una de ellas. Posteriormente, dedicamos una subsección a semiflujos discretos en la que recordamos las nociones de punto fijo, punto periódico, punto m -cíclico y m -ciclo, así como las de puntos finales y cuencas asociados a un semiflujo. Terminamos la sección comentando, para el caso de funciones racionales, las nociones de punto crítico y derivada esférica, diferenciando para puntos fijos y m -ciclos su posible carácter: super-atractor, atractor, indiferente o repulsor.

En la sección 2 mostramos una extensión canónica de los modelos S^2 y $\mathbf{P}^1(\mathbb{C})$, y analizamos la noción de coordenadas homogéneas normalizadas. Los nuevos modelos que utilizamos son: la 2-esfera aumentada S^{2+} y la recta proyectiva compleja ampliada $\mathbf{P}^{1+}(\mathbb{C})$. Estas estructuras nos permiten obtener una mejor representación de las funciones racionales, ya que posibilitan la obtención de la imagen en los casos en los que la función racional posee polos o incluso en aquellos casos en los que tanto numerador como denominador se anulan simultáneamente; y de esta forma, podemos evitar problemas de indefinición e indeterminación al realizar cálculos computacionales. Para ello, describimos el proceso de extensión de funciones racionales a estos nuevos modelos mediante el uso de pares homogéneos. Por último, comentamos el proceso de normalización para tales pares.

En la sección 3 presentamos y explicamos los algoritmos elaborados para la representación de cuencas de atracción de una función racional. Comenzamos describiendo el proceso de cálculo de puntos fijos de una función racional, distinguiendo los distintos casos que se pueden presentar. A continuación, comentamos una forma algorítmica de abordar el proceso de iteración de dichas funciones y, por último, exponemos los algoritmos utilizados para la construcción de entornos rectangulares, tanto del origen como del infinito. Estos entornos nos permiten visualizar globalmente todas las cuencas de atracción de los puntos fijos de la función racional.

En la sección 4 mostramos una colección de representaciones gráficas de conjuntos de Julia y conjuntos de Fatou (obtenidas con los algoritmos descritos en las dos últimas secciones), asociados a funciones polinómicas de segundo y tercer grado, propuestos por *John Milnor* en uno de sus artículos [10].

Estos conjuntos reciben sus nombres en honor a los matemáticos *Gaston Maurice Julia* (1893-1978) [7] y *Pierre Joseph Louis Fatou* (1878-1929) [5], respectivamente.

G. Julia nació en Argelia, bajo dominio colonial francés. A la edad de 21 años tuvo que interrumpir sus estudios debido al estallido de la Primera Guerra Mundial. Tras su intervención en el conflicto, pudo retomar sus investigaciones culminando con la publicación en 1918 de su libro “Mémoire sur l’itération des fonctions rationnelles” [7]. Sus descubrimientos le llevaron a ganar ese mismo año el “Grand Prix de l’Académie des Sciences”, en el que Fatou también participó. A partir de 1920, su trabajo fue esencialmente olvidado, hasta que en 1970 el matemático Benoît Mandelbrot lo hizo volver a resurgir con sus experimentos computacionales.



Figura 1:
Gaston Julia.

P. Fatou fue un matemático y astrónomo francés que escribió múltiples artículos matemáticos en temas variados de análisis y geometría. En 1906 comenzó a investigar sobre la iteración de funciones racionales y obtuvo los primeros resultados sobre este tema. Retomó la cuestión en 1917, tras la convocatoria del “Grand prix des Sciences mathématiques”. A pesar de que Fatou no lo ganó, continuó con sus investigaciones y en 1919 publicó sus principales resultados en el libro “Sur les équations fonctionnelles” [5].



Figura 2: Pierre Fatou.

Para cada una de las gráficas que mostramos en la sección 4, se realizan algunas consideraciones interesantes desde el punto de vista topológico con un análisis tanto del conjunto de Julia de una función racional como de sus puntos críticos, puntos fijos y ciclos; además, para estos dos últimos, analizamos su carácter en función del valor del multiplicador; esto es, si son super-atractores, atractores, indiferentes o repulsores, para lo cuál se utilizan los valores de las derivadas esféricas correspondientes.

Terminamos la sección con un caso particular de fractal denominado conjunto de Mandelbrot asociado a la familia de polinomios cuadráticos $f_c(z) = z^2 + c$ (algunos de los ejemplos de esta sección corresponden a valores concretos del parámetro c). Este conjunto recibe su nombre en honor al matemático *Benoît Mandelbrot* (1924-2010) [8]. También analizamos dicho conjunto y damos una representación gráfica del mismo obtenida mediante una implementación en Julia. Gracias a la visualización del conjunto, podemos observar los distintos lóbulos que presenta asociados con la convergencia a q -ciclos para distintos valores de q .

Los algoritmos utilizados para las representaciones gráficas contenidas en este TFG (de los que en algún caso se muestra su código) han sido implementados en el lenguaje de programación Julia. La razón por la que hemos elegido este entorno de programación radica en las ventajas que posee el mismo. Las principales características de este lenguaje son:

- i. Rapidez a la hora de realizar cálculos, la cual se aproxima (estadísticamente) a lenguajes como C.
- ii. Es posible utilizar las bibliotecas Python, C y Fortran. Julia puede interactuar directamente con bibliotecas externas escritas en C y Fortran, como por ejemplo la librería “Solvers” del lenguaje Python.
- iii. Dynamic typing: Se pueden especificar tipos para variables, como por ejemplo “real de 64 bits”, aunque también es posible prescindir de escribir completamente su tipo si no se conoce con exactitud el tipo que se va a utilizar en un contexto particular.
- iv. Está diseñado para computación paralela y distribuida.

Una de las características más notables de Julia es su compilador just-in-time (JIT), que permite disponer de versiones compiladas antes de ser ejecutadas. En la página oficial de Julia [18] podemos ver un “benchmark”, que compara a Julia con otros lenguajes como Fortran, Python, Java, Mathematica, R, Matlab, etc. (Puede verse más información de

Julia en [20]).

Estas propiedades del lenguaje de programación Julia, especialmente su rapidez en la ejecución, han sido utilizadas para realizar cada una de las representaciones gráficas que mostramos en el trabajo. En las correspondientes a las secciones 3 y 4 se ha utilizado como criterio para asignar un color a cada punto de la imagen la posición que ocupa un punto fijo, dentro de la lista de puntos fijos de la función racional, al cual converge (en caso de hacerlo) la secuencia de iteración de cada punto del entorno rectangular.

La representación mostrada en la portada de este trabajo se corresponde con el ejemplo de la subsección 4.9. En este caso, el criterio escogido para determinar el color de cada punto de la imagen ha sido el número de iteraciones necesarias para que la secuencia de iteración de esos puntos converja a uno de los puntos fijos de la función racional y los diferentes valores de posición han sido usados para cambiar las “intensidades” de los colores.

1. Preliminares

Comenzamos recordando algunas nociones y propiedades básicas sobre espacios topológicos (para una información más detallada sobre éstos y otros conceptos y resultados puede verse [17]).

Sea X un conjunto. Una *topología en X* es una familia \mathcal{T} de subconjuntos de X tal que \emptyset y X son elementos de \mathcal{T} y cualquier intersección finita o unión de elementos de \mathcal{T} es también un elemento de \mathcal{T} . Un *espacio topológico* es un par (X, \mathcal{T}) donde X es un conjunto y \mathcal{T} una topología en él. Por comodidad, si no hay lugar a confusión, suele denotarse el espacio topológico (X, \mathcal{T}) simplemente por X .

Un caso particular de espacios topológicos son los espacios topológicos *metrizables*: aquellos espacios (X, \mathcal{T}) donde la topología \mathcal{T} viene inducida por alguna métrica d de X (puede verse [14]).

Dados X e Y espacios topológicos, una aplicación $f : X \rightarrow Y$ es *continua* si para cada abierto V de Y se tiene que $f^{-1}(V)$ es abierto de X . Un *homeomorfismo* entre dos espacios topológicos X e Y es una aplicación $f : X \rightarrow Y$ biyectiva, continua y cuya inversa f^{-1} es continua. Se dice que dos espacios topológicos X e Y son *homeomorfos o topológicamente equivalentes* si existe un homeomorfismo entre ellos.

Recordamos, a continuación, algunas propiedades básicas de espacios topológicos:

Para un espacio topológico X , diremos que:

X es T_2 o de *Haussdorf* si verifica que cuando $x, y \in X$ con $x \neq y$, existen abiertos U y V disjuntos tales que $x \in U$ e $y \in V$.

X es *compacto* si todo cubrimiento abierto de X posee algún subcubrimiento finito; es decir, si dada cualquier familia \mathcal{U} de abiertos de X verificando que $X = \bigcup_{U \in \mathcal{U}} U$, existe una subfamilia finita $\mathcal{V} \subset \mathcal{U}$ tal que $X = \bigcup_{V \in \mathcal{V}} V$. X es *localmente compacto* si cada punto de X posee alguna base de entornos compactos.

X es *primero numerable* si cada punto de X posee una base numerable de entornos.

X es *segundo numerable* si su topología tiene alguna base numerable.

X es *conexo* si siempre que $X = U \cup V$ con U y V abiertos y $U \cap V = \emptyset$, entonces $U = \emptyset$ o $V = \emptyset$. X es *localmente conexo* si cada punto de X posee una base de entornos conexos.

X es *arcoconexo* si siempre que $x, y \in X$, existe una aplicación continua $f : [0, 1] \rightarrow X$

tal que $f(0) = x$ y $f(1) = y$. X es *localmente arcoconexo* si cada punto de X posee una base de entornos arcoconexos.

Se llama *componente conexa* o *componente* de X a todo subconjunto conexo maximal de X . Análogamente se definen las *componentes arcoconexas* o *arcocomponentes* de X .

1.1. Variedades topológicas

A continuación vamos a recordar algunas nociones y resultados sobre variedades topológicas (para más información sobre variedades puede verse [9]).

En lo que sigue, a no ser que se advierta lo contrario, \mathbb{R}^n denotará el espacio euclídeo n -dimensional con la métrica euclídea d dada por $d(x, y) = |x - y|$, inducida por la norma euclídea $|\cdot|$ ($|x| = \sqrt{\sum_{i=1}^n x_i^2}$, para cada $x = (x_1, \dots, x_n) \in \mathbb{R}^n$).

Definición 1.1.1. Sea n un entero positivo. Diremos que un espacio topológico M es una n -variedad topológica (o variedad topológica n -dimensional o, brevemente, n -variedad) si M es T_2 y cada $x \in M$ posee un entorno abierto U_x homeomorfo a un abierto de \mathbb{R}^n .

Proposición 1.1.1. Sea M un espacio topológico T_2 . Entonces son equivalentes:

- i. M es una n -variedad topológica.
- ii. $\forall x \in M$, existe un entorno abierto homeomorfo a la n -bola unidad $B^n = \{x \in \mathbb{R}^n \mid |x| < 1\}$.
- iii. Existe un cubrimiento abierto \mathcal{U} de M donde cada $U \in \mathcal{U}$ es homeomorfo a un abierto de \mathbb{R}^n .

Es muy fácil comprobar que \mathbb{R}^n y la n -esfera unidad $S^n = \{x \in \mathbb{R}^{n+1} \mid |x| = 1\}$ son n -variedades. Así mismo, si M es una n -variedad y U es un abierto de M , también U es una n -variedad.

Entre las diferentes propiedades que tiene una n -variedad topológica M , cabe destacar que es primero numerable, localmente compacta y localmente arcoconexa. Por tanto, las arcocomponentes de una n -variedad (que coinciden con sus componentes) son también n -variedades.

También es destacable que toda variedad topológica compacta (en cuyo caso sólo posee un número finito de componentes) puede incrustarse en algún espacio euclídeo \mathbb{R}^d , y, por tanto, es metrizable.

Definición 1.1.2. Una superficie es una 2-variedad topológica conexa y segundo numerable.

Definición 1.1.3. Sea M una n -variedad topológica. Una parametrización de M es una aplicación $\varphi : U \rightarrow M$ donde U es un abierto de \mathbb{R}^n y φ es una aplicación continua, inyectiva y abierta. A una familia de parametrizaciones de M , $\{\varphi_\lambda : U_\lambda \rightarrow M \mid \lambda \in \Lambda\}$, donde $M = \bigcup_{\lambda \in \Lambda} \varphi_\lambda(U_\lambda)$, la llamaremos cubrimiento parametrizador de M .

Notemos que cada aplicación $\varphi_\lambda : U_\lambda \rightarrow \varphi_\lambda(U_\lambda)$ es un homeomorfismo. A su inversa $\phi_\lambda = \varphi_\lambda^{-1} : \varphi_\lambda(U_\lambda) \rightarrow U_\lambda \subset \mathbb{R}^n$ se le llama *carta* de M y a la familia $\{\phi_\lambda \mid \lambda \in \Lambda\}$ se le llama *atlas topológico* de M .

En el caso de ser M una 2-variedad, si cambiamos \mathbb{R}^2 por el plano de los números complejos \mathbb{C} , utilizamos el lenguaje de *parametrizaciones complejas*.

1.2. Superficies de clase C^p y analíticas

Dada una 2-variedad topológica M , ésta posee un cubrimiento abierto \mathcal{V} tal que cada $V \in \mathcal{V}$ es homeomorfo a algún abierto euclídeo U de \mathbb{R}^2 . Si $\varphi : U \rightarrow V$ es un homeomorfismo y no hay lugar a confusión, a veces mantendremos la misma notación φ para su composición con la inclusión $\varphi : U \rightarrow V \hookrightarrow M$.

Seguidamente, vamos a recordar algunas nociones relacionadas con superficies; para más conceptos y resultados sobre superficies puede verse [3].

Definición 1.2.1. Sea M una 2-variedad topológica. Diremos que un cubrimiento parametrizador $\{\varphi_\lambda : U_\lambda \rightarrow M \mid \lambda \in \Lambda\}$ de M es de clase C^p si todos los cambios $\varphi_\mu^{-1}\varphi_\lambda$ son funciones de clase C^p . Si los cambios son funciones analíticas reales, diremos que se trata de un cubrimiento parametrizador analítico real. En el caso de tener un cubrimiento parametrizador complejo $\{\varphi_\lambda : U_\lambda \rightarrow M \mid \lambda \in \Lambda\}$ de M donde todos los cambios $\varphi_\mu^{-1}\varphi_\lambda$ son funciones analíticas complejas, diremos que se trata de un cubrimiento analítico complejo (1-dimensional) de M .

Definición 1.2.2. Sea M una 2-variedad topológica y sean $\mathcal{D}_1 = \{\varphi_\lambda : U_\lambda \rightarrow M \mid \lambda \in \Lambda\}$ y $\mathcal{D}_2 = \{\psi_\gamma : U_\gamma \rightarrow M \mid \gamma \in \Gamma\}$ dos cubrimientos parametrizadores de M . Si \mathcal{D}_1 y \mathcal{D}_2 son de clase C^p , diremos que son compatibles de clase C^p si todos los cambios $\psi_\gamma^{-1}\varphi_\lambda$, $\varphi_\lambda^{-1}\psi_\gamma$ son de clase C^p . De un modo similar se define la relación de compatibilidad en el caso de cubrimientos parametrizadores analíticos reales y analíticos complejos, respectivamente.

Definición 1.2.3. Sea M una 2-variedad topológica. Una estructura diferenciable de clase C^p sobre M es una clase de equivalencia de cubrimientos parametrizadores de clase C^p , módulo compatibilidad. De modo análogo tenemos el concepto de estructura analítica real y estructura analítica compleja sobre M .

Definición 1.2.4. Una 2-variedad de clase C^p es una 2-variedad topológica M junto con una estructura diferenciable de clase C^p sobre M . Análogamente, se definen las nociones de 2-variedad analítica real y 1-variedad analítica compleja.

Definición 1.2.5. Sea M una m -variedad y N una n -variedad y supongamos que $\{\varphi_\lambda : U_\lambda \rightarrow M \mid \lambda \in \Lambda\}$ y $\{\psi_\gamma : U_\gamma \rightarrow N \mid \gamma \in \Gamma\}$ son cubrimientos parametrizadores de clase C^p de M y N respectivamente. Diremos que una aplicación $f : M \rightarrow N$ es de clase C^p si para cada punto $x_0 \in M$ existen parametrizaciones $\varphi_\lambda, \psi_\gamma$ tales que x_0 y $f(x_0)$ están respectivamente en sus imágenes y la composición $\psi_\gamma^{-1}f\varphi_\lambda$ es una función de clase C^p . De manera similar, se definen los conceptos de aplicación analítica real y para el caso de que m y n sean enteros pares, el de aplicación analítica compleja.

Definición 1.2.6. Si M es una 2-variedad de clase C^p que es conexa y segundo numerable, diremos que M es una superficie de clase C^p . Análogamente, se tiene la noción de superficie analítica real. En el caso de que M sea una 1-variedad analítica compleja conexa y segundo numerable, diremos que M es una superficie de Riemann.

1.3. Ejemplos de Superficies de Riemann

Vamos a recordar que la esfera unidad 2-dimensional S^2 , el plano complejo ampliado $\hat{\mathbb{C}}$ y la recta proyectiva compleja $\mathbf{P}^1(\mathbb{C})$ son ejemplos de superficies de Riemann y a explicitar ciertas biyecciones canónicas que utilizaremos posteriormente.

1.3.1. La 2-esfera S^2

Consideramos $S^2 = \{(x_1, x_2, x_3) \in \mathbb{R}^3 \mid x_1^2 + x_2^2 + x_3^2 = 1\}$ con la topología usual. Sea $\varphi : \mathbb{C} \rightarrow S^2$ la aplicación dada por la proyección estereográfica determinada por el polo norte N de S^2 y el plano ecuatorial, cuya fórmula es:

$$\begin{aligned}\varphi(z) = \varphi(u + iv) &= \left(\frac{2u}{1+u^2+v^2}, \frac{2v}{1+u^2+v^2}, \frac{u^2+v^2-1}{1+u^2+v^2} \right) = \\ &= \left(\frac{2(z+\bar{z})/2}{1+z\bar{z}}, \frac{2(z-\bar{z})/(2i)}{1+z\bar{z}}, \frac{-1+z\bar{z}}{1+z\bar{z}} \right) = \left(\frac{z+\bar{z}}{1+z\bar{z}}, \frac{i(\bar{z}-z)}{1+z\bar{z}}, \frac{-1+z\bar{z}}{1+z\bar{z}} \right).\end{aligned}$$

Ahora, con la misma idea pero considerando el polo sur $S = -N$, podemos definir la aplicación $\phi : \mathbb{C} \rightarrow S^2$ dada por la composición de la proyección estereográfica y una reflexión respecto al plano ($x_2 = 0$):

$$\phi(z) = \phi(u + iv) = \left(\frac{2u}{1+u^2+v^2}, \frac{-2v}{1+u^2+v^2}, \frac{1-(u^2+v^2)}{1+u^2+v^2} \right) = \left(\frac{z+\bar{z}}{1+z\bar{z}}, \frac{-i(\bar{z}-z)}{1+z\bar{z}}, \frac{1-z\bar{z}}{1+z\bar{z}} \right).$$

Es fácil comprobar que $\{\varphi, \phi\}$ es un cubrimiento parametrizador de S^2 , y teniendo en cuenta que sus inversas vienen dadas por:

$$\varphi^{-1} : S^2 \setminus \{N\} \rightarrow \mathbb{C}: \varphi^{-1}((x_1, x_2, x_3)) = \frac{x_1}{1-x_3} + \frac{x_2}{1-x_3}i,$$

$$\phi^{-1} : S^2 \setminus \{S\} \rightarrow \mathbb{C}: \phi^{-1}((x_1, x_2, x_3)) = \frac{x_1}{1+x_3} - \frac{x_2}{1+x_3}i,$$

se obtiene que los cambios de variable vienen dados por:

$$\varphi^{-1}(\phi(z)) = \varphi^{-1}(\phi(u + iv)) = \varphi^{-1} \left(\frac{2u}{1+u^2+v^2}, \frac{-2v}{1+u^2+v^2}, \frac{1-u^2-v^2}{1+u^2+v^2} \right) = \frac{u}{u^2+v^2} - \frac{v}{u^2+v^2}i = \frac{1}{z},$$

$$\phi^{-1}(\varphi(z)) = \phi^{-1}(\varphi(u + iv)) = \phi^{-1} \left(\frac{2u}{1+u^2+v^2}, \frac{2v}{1+u^2+v^2}, \frac{u^2+v^2-1}{1+u^2+v^2} \right) = \frac{u}{u^2+v^2} - \frac{v}{u^2+v^2}i = \frac{1}{z},$$

y como $\frac{1}{z}$ es una función analítica compleja, también los son las funciones $\varphi^{-1}\phi$, $\phi^{-1}\varphi : \mathbb{C} \rightarrow \mathbb{C}$.

Por tanto, S^2 admite una estructura de 1-variedad analítica compleja con el cubrimiento parametrizador $\{\varphi, \phi\}$ y, como es bien conocido que S^2 es conexa y segundo numerable, es una superficie de Riemann.

1.3.2. El plano complejo ampliado $\hat{\mathbb{C}}$

Se define el plano complejo ampliado $\hat{\mathbb{C}}$ como la compactificación de Alexandroff de \mathbb{C} ; es decir, $\hat{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$ (donde ∞ es un punto tal que $\infty \notin \mathbb{C}$) con la topología de Alexandroff: $\mathcal{T}^* = \{U \subset \mathbb{C} \mid U \text{ es abierto de } \mathbb{C}\} \cup \{\{\infty\} \cup (\mathbb{C} \setminus K) \mid K \subset \mathbb{C} \text{ con } K \text{ compacto}\}$.

Consideremos las aplicaciones $in, inv : \mathbb{C} \rightarrow \hat{\mathbb{C}}$ dadas por la inclusión y la inversión respectivamente:

$$in(z) = z, \forall z \in \mathbb{C} \quad \text{y} \quad inv(z) = \begin{cases} \frac{1}{z}, & \text{si } z \neq 0 \\ \infty, & \text{si } z = 0. \end{cases}$$

Es fácil comprobar que $\{in, inv\}$ es un cubrimiento parametrizador de $\hat{\mathbb{C}}$ teniendo en cuenta que sus inversas vienen dadas por:

$$in^{-1} : \hat{\mathbb{C}} \setminus \{\infty\} \rightarrow \mathbb{C} : in^{-1}(z) = z, \forall z \in \hat{\mathbb{C}} \setminus \{\infty\}.$$

$$inv^{-1} : \hat{\mathbb{C}} \setminus \{0\} \rightarrow \mathbb{C} : inv^{-1}(z) = \begin{cases} \frac{1}{z}, & \text{si } z \neq \infty \\ 0, & \text{si } z = \infty. \end{cases}$$

Puesto que $in^{-1}(inv(z)) = \frac{1}{z}$, $inv^{-1}(in(z)) = \frac{1}{z}$ y $\frac{1}{z}$ es una función analítica compleja, las composiciones $in^{-1}inv$ e $inv^{-1}in$ también lo son.

Por tanto, $\hat{\mathbb{C}}$ admite una estructura de 1-variedad analítica compleja con el cubrimiento parametrizador $\{in, inv\}$ y, como $\hat{\mathbb{C}}$ es conexo y segundo numerable, es una superficie de Riemann. De hecho, al plano complejo ampliado $\hat{\mathbb{C}}$ se le llama en numerosas ocasiones Esfera de Riemann.

1.3.3. La recta proyectiva compleja $\mathbf{P}^1(\mathbb{C})$

Se define la recta proyectiva compleja $\mathbf{P}^1(\mathbb{C})$ como el espacio cociente

$$\mathbf{P}^1(\mathbb{C}) = \frac{\mathbb{C}^2 \setminus \{(0, 0)\}}{\sim},$$

donde $(z, t) \sim (z', t')$ si $\exists \lambda \in \mathbb{C} \setminus \{0\}$ tal que $(z, t) = (\lambda z', \lambda t')$.

Denotamos por $[z : t] \in \mathbf{P}^1(\mathbb{C})$ a la clase de equivalencia de $(z, t) \in \mathbb{C}^2 \setminus \{(0, 0)\}$.

Definición 1.3.1. Dado un punto $[z : t] \in \mathbf{P}^1(\mathbb{C})$, se dice que (z, t) son las coordenadas homogéneas del punto y t/z (o z/t en su caso) se dice que es la coordenada absoluta de ese punto.

Consideremos las aplicaciones $\psi, \nu : \mathbb{C} \rightarrow \mathbf{P}^1(\mathbb{C})$ definidas por:

$$\psi(z) = [1 : z] \text{ y } \nu(z) = [z : 1].$$

Es fácil comprobar que $\{\psi, \nu\}$ es un cubrimiento parametrizador de $\mathbf{P}^1(\mathbb{C})$ teniendo en cuenta que sus inversas vienen dadas por:

$$\begin{aligned} \psi^{-1} : \mathbf{P}^1(\mathbb{C}) \setminus \{[0 : 1]\} &\rightarrow \mathbb{C}, \quad \psi^{-1}([z : t]) = \frac{t}{z}. \\ \nu^{-1} : \mathbf{P}^1(\mathbb{C}) \setminus \{[1 : 0]\} &\rightarrow \mathbb{C}, \quad \nu^{-1}([z : t]) = \frac{z}{t}. \end{aligned}$$

Dado que $\nu^{-1}(\psi(z)) = \nu^{-1}([1 : z]) = \frac{1}{z}$, $\psi^{-1}(\nu(z)) = \psi^{-1}([z : 1]) = \frac{1}{z}$ y $\frac{1}{z}$ es una función analítica compleja, las composiciones $\nu^{-1}\psi$ y $\psi^{-1}\nu$ son funciones analíticas complejas.

Por tanto, $\mathbf{P}^1(\mathbb{C})$ admite una estructura de 1-variedad analítica compleja con la parametrización $\{\psi, \nu\}$ y, puesto que es conexa y segundo numerable, es una superficie de Riemann.

1.3.4. Biyecciones canónicas

A continuación, vamos a analizar la relación de las estructuras descritas anteriormente. Para ello, consideramos el siguiente diagrama:

$$\begin{array}{ccc} & \hat{\mathbb{C}} & \\ \theta \nearrow & & \nwarrow \tilde{\theta} \\ \mathbf{P}^1(\mathbb{C}) & \xrightarrow{\tilde{\theta}^{-1}\theta} & S^2 \end{array}$$

donde $\theta : \mathbf{P}^1(\mathbb{C}) \rightarrow \hat{\mathbb{C}}$ y $\tilde{\theta} : S^2 \rightarrow \hat{\mathbb{C}}$ son las biyecciones definidas por:

$$\theta([z : t]) = \begin{cases} \frac{z}{t}, & \text{si } t \neq 0 \\ \infty, & \text{si } t = 0 \end{cases} \quad \text{y} \quad \tilde{\theta}((x_1, x_2, x_3)) = \begin{cases} \frac{x_1}{1-x_3} + \frac{x_2}{1-x_3}i, & \text{si } x_3 < 1 \\ \infty, & \text{si } x_3 = 1. \end{cases}$$

Notemos que sus inversas $\theta^{-1} : \hat{\mathbb{C}} \rightarrow \mathbf{P}^1(\mathbb{C})$ y $\tilde{\theta}^{-1} : \hat{\mathbb{C}} \rightarrow S^2$ vienen dadas por:

$$\begin{aligned} \theta^{-1}(z) &= \begin{cases} [z : 1], & \text{si } z \neq \infty \\ [z : 0], & \text{si } z = \infty, \end{cases} \\ \tilde{\theta}^{-1}(z) = \tilde{\theta}^{-1}(u+iv) &= \begin{cases} \left(\frac{2u}{1+u^2+v^2}, \frac{2v}{1+u^2+v^2}, \frac{1-(u^2+v^2)}{1+u^2+v^2} \right) = \left(\frac{z+\bar{z}}{1+z\bar{z}}, \frac{i(\bar{z}-z)}{1+z\bar{z}}, \frac{-1+z\bar{z}}{1+z\bar{z}} \right), & \text{si } z \neq \infty \\ (0, 0, 1), & \text{si } z = \infty. \end{cases} \end{aligned}$$

Luego $\tilde{\theta}^{-1}\theta : \mathbf{P}^1(\mathbb{C}) \rightarrow S^2$ viene dada por:

$$(\tilde{\theta}^{-1}\theta)([z : t]) = \begin{cases} \tilde{\theta}^{-1}\left(\frac{z}{t}\right), & \text{si } t \neq 0, \\ \tilde{\theta}^{-1}(\infty), & \text{si } t = 0 \end{cases} = \begin{cases} \left(\frac{z\bar{t}+t\bar{z}}{t\bar{t}+z\bar{z}}, \frac{i(t\bar{z}-z\bar{t})}{t\bar{t}+z\bar{z}}, \frac{-t\bar{t}+z\bar{z}}{t\bar{t}+z\bar{z}} \right), & \text{si } t \neq 0 \\ (0, 0, 1), & \text{si } t = 0. \end{cases}$$

Lo que se puede expresar conjuntamente con la fórmula:

$$(\tilde{\theta}^{-1}\theta)([z : t]) = \left(\frac{z\bar{t} + t\bar{z}}{t\bar{t} + z\bar{z}}, \frac{i(t\bar{z} - z\bar{t})}{t\bar{t} + z\bar{z}}, \frac{-t\bar{t} + z\bar{z}}{t\bar{t} + z\bar{z}} \right).$$

La inversa de la biyección anterior, $\theta^{-1}\tilde{\theta} : S^2 \rightarrow \mathbf{P}^1(\mathbb{C})$ viene dada por la siguiente fórmula:

$$(\theta^{-1}\tilde{\theta})((x_1, x_2, x_3)) = \begin{cases} \theta^{-1}\left(\frac{x_1}{1-x_3} + i\frac{x_2}{1-x_3}\right), & \text{si } x_3 < 1 \\ \theta^{-1}(\infty), & \text{si } x_3 = 1 \end{cases} = \begin{cases} [x_1 + i x_2 : 1 - x_3], & \text{si } x_3 < 1 \\ [x_1 + i x_2 : 0], & \text{si } x_3 = 1. \end{cases}$$

O equivalentemente, por:

$$(\theta^{-1}\tilde{\theta})((x_1, x_2, x_3)) = [x_1 + i x_2 : 1 - x_3].$$

No es difícil comprobar que las biyecciones θ y $\tilde{\theta}$ son analíticas complejas. En el caso de θ , dado que tanto $\mathbf{P}^1(\mathbb{C})$ como $\hat{\mathbb{C}}$ son 1-variedades analíticas complejas, podemos considerar el siguiente diagrama

$$\begin{array}{ccc} \mathbf{P}^1(\mathbb{C}) & \xrightarrow{\theta} & \hat{\mathbb{C}} \\ \uparrow \{\psi, \nu\} & & \uparrow \{in, inv\} \\ \mathbb{C} & & \mathbb{C} \end{array}$$

y comprobar que las correspondientes composiciones (ver Definición 1.2.5) son funciones analíticas complejas. Por ejemplo, $inv^{-1} \theta \psi : \mathbb{C} \rightarrow \mathbb{C}$ viene dada por:

$$inv^{-1}(\theta(\psi(z))) = inv^{-1}(\theta([1 : z])) = \begin{cases} inv^{-1}(\frac{1}{z}), & \text{si } z \neq 0 \\ inv^{-1}(\infty), & \text{si } z = 0 \end{cases} = \begin{cases} z, & \text{si } z \neq 0 \\ 0, & \text{si } z = 0 \end{cases} = z.$$

y obviamente la identidad es analítica compleja.

Para el caso de $\tilde{\theta}$, sabiendo que S^2 y $\hat{\mathbb{C}}$ son 1-variedades analíticas complejas, podemos considerar el diagrama:

$$\begin{array}{ccc} S^2 & \xrightarrow{\tilde{\theta}} & \hat{\mathbb{C}} \\ \uparrow \{\varphi, \phi\} & & \uparrow \{in, inv\} \\ \mathbb{C} & & \mathbb{C} \end{array}$$

y comprobar que las correspondientes composiciones son analíticas complejas. Por ejemplo, $in^{-1} \tilde{\theta} \varphi : \mathbb{C} \rightarrow \mathbb{C}$ es una aplicación analítica compleja, pues:

$$in^{-1}(\tilde{\theta}(\varphi(z))) = in^{-1}\left(\tilde{\theta}\left(\frac{z + \bar{z}}{1 + z\bar{z}}, \frac{i(\bar{z} - z)}{1 + z\bar{z}}, \frac{-1 + z\bar{z}}{1 + z\bar{z}}\right)\right) = in^{-1}\left(\frac{z + \bar{z}}{2} + \frac{z - \bar{z}}{2i}i\right) = z.$$

Puesto que las biyecciones anteriores $\mathbf{P}^1(\mathbb{C}) \cong \hat{\mathbb{C}} \cong S^2$ son analíticas complejas, a cualquiera de estas superficies se le llama Esfera de Riemann.

1.3.5. Métricas cordales

Describimos las métricas que consideramos en S^2 , $\hat{\mathbb{C}}$ y $\mathbf{P}^1(\mathbb{C})$:

Dado que $S^2 \subset \mathbb{R}^3$, la métrica Euclídea usual de \mathbb{R}^3 induce una métrica cordal Euclídea en S^2 : Si $p = (x_1, x_2, x_3)$, $p' = (x'_1, x'_2, x'_3) \in S^2$,
 $d(p, p') = \sqrt{(x_1 - x'_1)^2 + (x_2 - x'_2)^2 + (x_3 - x'_3)^2}$, pero como $p, p' \in S^2$, entonces: $d(p, p') = \sqrt{2 - 2(x_1x'_1 + x_2x'_2 + x_3x'_3)}$.

Usando la biyección $\tilde{\theta}^{-1} : \hat{\mathbb{C}} \rightarrow S^2$, trasladamos esta estructura métrica en S^2 a $\hat{\mathbb{C}}$ mediante la fórmula:

$$d_1(z, z') = d(\tilde{\theta}^{-1}(z), \tilde{\theta}^{-1}(z')).$$

y a través de la biyección $\tilde{\theta}^{-1}\theta : \mathbf{P}^1(\mathbb{C}) \rightarrow S^2$, podemos trasladar la misma estructura métrica en S^2 a $\mathbf{P}^1(\mathbb{C})$, expresada con la fórmula:

$$d_2([z : t], [z' : t']) = d((\tilde{\theta}^{-1}\theta)([z : t]), (\tilde{\theta}^{-1}\theta)([z' : t'])).$$

En lo que sigue, muchas veces a todas estas métricas las llamaremos métrica cordal y para simplificar, a todas ellas las denotaremos en el modelo correspondiente como d .

1.4. Semiflujos discretos

En esta subsección introduciremos algunas definiciones básicas sobre semiflujos discretos.

J. H. Poincaré (1854-1912) [13] fue un matemático francés pionero en el estudio de las propiedades topológicas de soluciones de ecuaciones diferenciales ordinarias. Muchos de sus trabajos estudiaron las propiedades globales de los sistemas dinámicos (flujos). La teoría de los sistemas dinámicos alcanzó un gran desarrollo gracias al trabajo del matemático estadounidense G. D. Birkhoff (1884-1944) [2], al cual suele considerarse el fundador de esta teoría. Estableció dos ramas principales en el estudio de los sistemas dinámicos: la teoría topológica y la teoría ergódica.

1.4.1. Semiflujos discretos

En lo que sigue supondremos $\mathbb{N} = \{0, 1, 2, \dots\}$ dotado de la topología usual; es decir, la topología discreta $\mathcal{P}(\mathbb{N})$.

Definición 1.4.1. *Un semiflujo discreto es un par (X, ϕ) , donde X es un espacio topológico y $\phi : \mathbb{N} \times X \rightarrow X$ es una aplicación continua tal que:*

- i. $\phi(0, x) = x, \forall x \in X$.

$$ii. \phi(m, \phi(n, x)) = \phi(m + n, x), \forall x \in X, \forall n, m \in \mathbb{N}.$$

Cuando no haya lugar a confusión, usaremos únicamente X para denotar al semiflujo discreto (X, ϕ) . Además, es común utilizar la notación $\phi(n, x) = n \cdot x$ para abreviar.

Definición 1.4.2. Si (X, ϕ) e (Y, ψ) son dos semiflujos discretos, un morfismo de semiflujos discretos $\eta : (X, \phi) \rightarrow (Y, \psi)$ es una aplicación continua $\eta : X \rightarrow Y$ tal que $\eta(n \cdot x) = n \cdot \eta(x)$, $\forall x \in X, \forall n \in \mathbb{N}$.

Si (X, ϕ) es un semiflujo discreto, dados $n_0 \in \mathbb{N}$ y $x_0 \in X$, se tienen las siguientes aplicaciones continuas inducidas: $\phi^{n_0} : X \rightarrow X$ definida por $\phi^{n_0}(x) = \phi(n_0, x)$, $\forall x \in X$, y la trayectoria de x_0 , $\phi_{x_0} : \mathbb{N} \rightarrow X$ dada por $\phi_{x_0}(n) = \phi(n, x_0)$, $\forall n \in \mathbb{N}$. A la sucesión $(\phi_{x_0}(n))_{n \in \mathbb{N}}$ se le llama *órbita de x_0* .

En particular, es interesante el caso $n_0 = 1$. En efecto, un semiflujo discreto $\phi : \mathbb{N} \times X \rightarrow X$ induce una aplicación continua $\phi^1 : X \rightarrow X$ dada por $\phi^1(x) = \phi(1, x) = 1 \cdot x$, $\forall x \in X$ y, recíprocamente, una aplicación continua $f : X \rightarrow X$ induce un semiflujo discreto $\phi : \mathbb{N} \times X \rightarrow X$ dado por la iteración de f : $\phi(n, x) = f^n(x)$, donde $f^0 = id_X$ y para $n \neq 0$, f^n denota la composición $f \circ \dots \circ f$.

Por tanto, es equivalente tener un semiflujo discreto expresado como un par (X, ϕ) verificando las condiciones de la Definición 1.4.1 que tener un par (X, ϕ^1) donde $\phi^1 : X \rightarrow X$ es una aplicación continua.

Definición 1.4.3. Sea (X, ϕ) un semiflujo discreto y $x \in X$. Diremos que x es un punto fijo si $\forall n \in \mathbb{N}, \phi^n(x) = x$. Se dirá que x es un punto periódico si $\exists n \in \mathbb{N}, n \neq 0$, tal que $\phi^n(x) = x$, y que x es un punto m -cíclico si $\phi^m(x) = x$ y además $\{x, \phi(x), \dots, \phi^{m-1}(x)\}$ tiene m elementos.

Si x es un punto m -cíclico, diremos que $(x, \phi(x), \dots, \phi^{m-1}(x))$ es un m -ciclo de ϕ .

Definición 1.4.4. Supongamos que X es un espacio métrico con métrica d y que tenemos un semiflujo discreto (X, f) dado por una aplicación continua $f : X \rightarrow X$. Consideramos la siguiente relación de equivalencia (\sim) en X :

$$x \sim x' \text{ si y sólo si } d(f^n(x), f^n(x')) \xrightarrow{n \rightarrow +\infty} 0.$$

Entonces se tiene el cociente $\frac{X}{\sim}$ y la aplicación natural $q : X \rightarrow \frac{X}{\sim}$. A cada clase de equivalencia $[x_0] \in \frac{X}{\sim}$ la llamaremos punto final del semiflujo discreto $X = (X, f)$.

Dado un punto m -cíclico $x_0 \in X$, se llamará cuenca de x_0 a $q^{-1}([x_0])$ y cuenca del m -ciclo $C = (x_0, f(x_0), \dots, f^{m-1}(x_0))$ a $\bigcup_{i=0}^{m-1} q^{-1}([f^i(x_0)])$.

1.4.2. Funciones racionales complejas

En este apartado vamos a centrar el estudio de funciones racionales complejas y nociones básicas asociadas utilizando uno de los modelos descritos anteriormente. Para un

estudio más detallado de estas funciones puede verse [1].

Sean $\mathbb{C}[u]$ el anillo de los polinomios con coeficientes complejos y $\mathbb{C}(u)$ el cuerpo de las funciones racionales con coeficientes complejos. Todo elemento $f \in \mathbb{C}(u)$ puede ser representado de la siguiente forma: $f = \frac{A}{B}$ donde A y B son polinomios en la variable u de grados n y m respectivamente, tales que B es no nulo.

Si $A(u) = a_0 + a_1u + \cdots + a_nu^n$ y $B(u) = b_0 + b_1u + \cdots + b_mu^m$, y suponemos que $\frac{A}{B}$ es una representación irreducible de f ; es decir, los polinomios A y B no tienen raíces comunes, entonces $f \in \mathbb{C}(u)$ determina una función $f : \mathbb{C} \rightarrow \mathbb{C}$, dada por:

$$f(u) = \frac{A(u)}{B(u)} = \frac{a_0 + a_1u + \cdots + a_nu^n}{b_0 + b_1u + \cdots + b_mu^m}. \quad (1)$$

Si $B(u) \neq 0$, entonces $f(u) \in \mathbb{C}$. En otro caso, $f(u)$ no está definido. Como $\frac{A}{B}$ es una representación irreducible de f , puede considerarse una aplicación extensión de f a $\hat{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$, que denotaremos por $\hat{f} : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$, dada por:

$$\hat{f}(u) = \begin{cases} \frac{A(u)}{B(u)}, & \text{si } B(u) \neq 0 \\ \infty, & \text{si } B(u) = 0 \end{cases} \quad \text{y} \quad \hat{f}(\infty) = \begin{cases} 0 & \text{si } n < m \\ \infty & \text{si } n > m \\ \frac{a_n}{b_m} & \text{si } n = m. \end{cases} \quad (2)$$

En muchas ocasiones, en lugar de denotar \hat{f} , se indica lo anterior diciendo que se considera la aplicación racional $f : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$. La iteración de tal función racional determina por tanto un semiflujo discreto en $\hat{\mathbb{C}}$.

Definición 1.4.5. Sea $f : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$ una aplicación racional. Se denomina conjunto de Fatou, y se denota por $F(f)$, al conjunto formado por los puntos $z_0 \in \hat{\mathbb{C}}$ verificando que $\forall \epsilon > 0, \exists \delta > 0$ tal que si $d(z, z_0) < \delta$, $z \in \hat{\mathbb{C}}$, entonces $d(f^n(z), f^n(z_0)) < \epsilon$, $\forall n \geq 0$.

Al complementario en $\hat{\mathbb{C}}$ del conjunto de Fatou, $\hat{\mathbb{C}} \setminus F(f)$, se denomina conjunto de Julia de f , y se denota por $J(f)$.

Notemos que el conjunto de Fatou $F(f)$ es un subconjunto abierto de $\hat{\mathbb{C}}$. Así que el conjunto de Julia $J(f)$ es un subconjunto cerrado de $\hat{\mathbb{C}}$ y, por lo tanto, $J(f)$ es compacto.

Definición 1.4.6. Sea $f : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$ una aplicación racional. Diremos que $z_0 \in \hat{\mathbb{C}}$ es un punto crítico de f si existen parametrizaciones $\varphi, \phi \in \{in, inv\}$ en z_0 y $f(z_0)$ respectivamente tales que $(\phi^{-1}f\varphi)'(\varphi^{-1}(z_0)) = 0$. Al valor $f(z_0)$ se le llama valor crítico asociado.

Definición 1.4.7. Dada una aplicación racional $f : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$, se llama derivada esférica de f a la aplicación $f^\sharp : \hat{\mathbb{C}} \rightarrow [0, +\infty]$ definida por la expresión:

$$f^\sharp(z_0) = |f'(z_0)| \frac{1 + |z_0|^2}{1 + |f(z_0)|^2},$$

donde $f'(z_0)$ denota la derivada de f en los puntos $z_0 \in \mathbb{C}$ para los que f' está definida y, en otro caso, se obtiene $f^\sharp(z_0)$ como el límite cuando $z \rightarrow z_0$ de toda la expresión de la derecha.

Se puede probar que el límite anterior siempre existe y, en consecuencia, la función $f^\#$ está bien definida. Además la función $f^\# : \hat{\mathbb{C}} \rightarrow [0, +\infty)$ es continua y por ser $\hat{\mathbb{C}}$ compacto es también acotada.

Observemos que si $z_0 \in \mathbb{C}$ es un punto fijo de f y f' está definida en z_0 , entonces $f^\#(z_0) = |f'(z_0)|$. Notemos también que un punto $z_0 \in \hat{\mathbb{C}}$ es un punto crítico de f si y solo si $f^\#(z_0) = 0$.

Definición 1.4.8. Dada una aplicación racional $f : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$ y $z_0 \in \hat{\mathbb{C}}$ un punto fijo de f , diremos que z_0 es super-atractor si $f^\#(z_0) = 0$, atractor si $f^\#(z_0) < 1$, indiferente si $f^\#(z_0) = 1$ y repulsor si $f^\#(z_0) > 1$.

Notemos que todo punto fijo super-atractor es un punto crítico de f .

Definición 1.4.9. Dada una aplicación racional $f : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$, sea $C = (z_0, f(z_0), \dots, f^{m-1}(z_0))$ un m -ciclo de f . Llamamos multiplicador del m -ciclo C al producto:

$$\mu(C) = f^\#(z_0) \cdot f^\#(f(z_0)) \cdots f^\#(f^{m-1}(z_0)).$$

Diremos que el m -ciclo C es super-atractor si $\mu(C) = 0$, atractor si $\mu(C) < 1$, indiferente si $\mu(C) = 1$ y repulsor si $\mu(C) > 1$.

Proposición 1.4.1. Sean $f : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$ una aplicación racional, $z_0 \in \mathbb{C}$ y $C = (z_0, f(z_0), \dots, f^{m-1}(z_0))$ un m -ciclo de f . En el caso en el que las derivadas que aparecen en las expresiones siguientes estén definidas, se tiene:

- i. $\mu(C) = |f'(z_0)| \cdots |f'(f^{m-1}(z_0))|$;
- ii. $\mu(C) = (f^m)^\#(f^k(z_0)) = |(f^m)'(f^k(z_0))|$, para cada k con $0 \leq k \leq m-1$.

2. Extensión de modelos y pares homogéneos

En las dos secciones siguientes, vamos a comentar los modelos matemáticos propuestos en [4] para la representación gráfica de cuencas de atracción asociadas a funciones racionales. Además, mostraremos los algoritmos usados e incluiremos algunas de las mejoras específicas que hemos realizado en este TFG para la utilización de programación en paralelo.

2.1. Modelos extendidos

Para el desarrollo de nuestros algoritmos vamos a introducir unas extensiones de la 2-esfera S^2 y la recta proyectiva compleja $\mathbf{P}^1(\mathbb{C})$.

Cuando estamos trabajando en un entorno informático, desde un punto de vista computacional, necesitamos prefijar cierta tolerancia que nos permita establecer cuando un número es lo suficientemente pequeño como para tomarlo como el cero. Así, cuando (z, t) son las coordenadas homogéneas de un punto $[z : t] \in \mathbf{P}^1(\mathbb{C})$ y se verifica que $|z| + |t|$ es menor que un cierto valor prefijado, entonces, computacionalmente, se interpretará que (z, t) es $(0, 0)$.

Puesto que $(0, 0)$ no son las coordenadas homogéneas de ningún punto de $\mathbf{P}^1(\mathbb{C})$ es por lo que resulta conveniente ampliar el modelo $\mathbf{P}^1(\mathbb{C})$ con un punto extra que llamaremos super-cero y lo denotaremos por $\bar{0} = [0 : 0]$. De este modo, consideramos el modelo $\mathbf{P}^{1+}(\mathbb{C})$ al que llamaremos recta proyectiva compleja ampliada:

$$\mathbf{P}^{1+}(\mathbb{C}) = \mathbf{P}^1(\mathbb{C}) \sqcup \{[0 : 0]\}.$$

Notemos que $\mathbf{P}^{1+}(\mathbb{C})$ puede pensarse como el cociente $\frac{\mathbb{C}^2}{\sim}$, donde $(z, t) \sim (z', t')$ si $\exists \lambda \in \mathbb{C} \setminus \{0\}$ tal que $(z, t) = (\lambda z', \lambda t')$.

También consideramos la 2-esfera aumentada S^{2+} dada por:

$$S^{2+} = S^2 \sqcup \{(0, 0, 0)\}.$$

Las biyecciones descritas en la sección anterior pueden extenderse a biyecciones entre los modelos ampliados: $(\tilde{\theta}^{-1}\theta)^+ : \mathbf{P}^{1+}(\mathbb{C}) \rightarrow S^{2+}$ y su inversa $(\theta^{-1}\tilde{\theta})^+ : \mathbf{P}^{1+}(\mathbb{C}) \rightarrow S^{2+}$, dadas por:

$$\begin{aligned}
(\tilde{\theta}^{-1}\theta)^+([z : t]) &= \begin{cases} \left(\frac{z\bar{t}+t\bar{z}}{t\bar{t}+z\bar{z}}, \frac{i(t\bar{z}-z\bar{t})}{t\bar{t}+z\bar{z}}, \frac{-t\bar{t}+z\bar{z}}{t\bar{t}+z\bar{z}} \right), & \text{si } t\bar{t} + z\bar{z} \neq 0 \\ (0, 0, 0), & \text{si } t\bar{t} + z\bar{z} = 0, \end{cases} \\
(\theta^{-1}\tilde{\theta})^+((x_1, x_2, x_3)) &= \begin{cases} [x_1 + i x_2 : 1 - x_3], & \text{si } (x_1, x_2, x_3) \neq (0, 0, 0) \\ [0 : 0], & \text{si } (x_1, x_2, x_3) = (0, 0, 0). \end{cases}
\end{aligned}$$

Notemos que esta última puede expresarse con la fórmula:

$$(\theta^{-1}\tilde{\theta})^+((x_1, x_2, x_3)) = [x_1 + i x_2 : 1 - x_3].$$

Para poder implementar en el lenguaje de programación Julia las biyecciones anteriores, hemos considerado la aplicación **sphereBijection**: $\mathbb{C}^2 \rightarrow S^{2+}$ dada por:

$$\text{sphereBijection}((z, t)) = \begin{cases} \left(\frac{z\bar{t}+t\bar{z}}{t\bar{t}+z\bar{z}}, \frac{i(t\bar{z}-z\bar{t})}{t\bar{t}+z\bar{z}}, \frac{-t\bar{t}+z\bar{z}}{t\bar{t}+z\bar{z}} \right), & \text{si } t\bar{t} + z\bar{z} \neq 0 \\ (0, 0, 0), & \text{si } t\bar{t} + z\bar{z} = 0. \end{cases} \quad (3)$$

Esta aplicación permite obtener las coordenadas espaciales de un punto de $\mathbf{P}^{1+}(\mathbb{C})$ con coordenadas homogéneas (z, t) .

Podemos extender las métricas descritas en 1.3.5 a estas nuevas estructuras. Utilizando el hecho de que $S^{2+} \subset \mathbb{R}^3$, la métrica Euclídea usual de \mathbb{R}^3 induce una métrica cordal Euclídea d en S^{2+} que extiende la considerada en S^2 .

Tomando la biyección $(\tilde{\theta}^{-1}\theta)^+ : \mathbf{P}^{1+}(\mathbb{C}) \rightarrow S^{2+}$, podemos trasladar esta estructura métrica en S^{2+} a $\mathbf{P}^{1+}(\mathbb{C})$ mediante la fórmula:

$$d_2^+([z : t], [z' : t']) = d((\tilde{\theta}^{-1}\theta)^+([z : t]), (\tilde{\theta}^{-1}\theta)^+([z' : t'])).$$

Para implementar dicha métrica en Julia, consideramos la aplicación **chordalMetric**: $\mathbb{C}^2 \times \mathbb{C}^2 \rightarrow [0, 2]$ dada por:

$$\text{chordalMetric}((z, t), (z', t')) = d(\text{sphereBijection}((z, t)), \text{sphereBijection}((z', t'))). \quad (4)$$

Muchas veces a todas estas métricas las llamaremos métrica cordal y, para simplificar, a todas ellas las denotaremos en el modelo correspondiente como d .

2.2. Coordenadas homogéneas normalizadas

Con frecuencia, al realizar ciertos cálculos computacionales pueden aparecer números muy grandes o muy pequeños que hacen que los sistemas informáticos cometan errores o incluso no puedan operar con ellos. En nuestro contexto, con el fin de evitar dichos problemas, trabajaremos con coordenadas homogéneas normalizadas.

Definición 2.2.1. Sean (z, t) las coordenadas homogéneas de un punto $[z : t] \in \mathbf{P}^{1+}(\mathbb{C})$. Llamaremos coordenadas homogéneas normalizadas al par: $\left(\frac{z}{|z|+|t|}, \frac{t}{|z|+|t|}\right)$ si $[z : t] \neq [0 : 0]$ y $(0, 0)$ si $[z : t] = [0 : 0]$.

Notar que si (z, t) son las coordenadas homogéneas del punto $[z : t]$, entonces las coordenadas homogéneas normalizadas (z', t') descritas en la definición anterior representan también a dicho punto.

En cada entorno computacional fijaremos una tolerancia de 10^{-N} , de manera que cuando un punto $[z : t] \in \mathbf{P}^1(\mathbb{C})$ con coordenadas homogéneas (z, t) , verifique que $|z| + |t| < 10^{-N}$, entonces diremos que (z, t) es computacionalmente interpretado como $(0, 0)$. Al valor 10^{-N} lo llamaremos Tolerancia de Nulidad.

Así, podemos definir una función en Julia que calcule las coordenadas homogéneas normalizadas de un par (z, t) fijando una cierta Tolerancia de Nulidad. Dicha función está implementada en Julia mediante la aplicación

$$\text{homogeneousNormalization}: \mathbb{C}^2 \times \mathbb{N} \rightarrow D \times D$$

dada por:

$$\text{homogeneousNormalization}((z, t), N) = \begin{cases} (0, 0), & \text{si } |z| + |t| < 10^{-N} \\ \left(\frac{z}{|z|+|t|}, \frac{t}{|z|+|t|}\right), & \text{en otro caso.} \end{cases} \quad (5)$$

2.3. Aplicaciones racionales complejas y pares homogéneos

En esta subsección vamos a estudiar la representación de una función racional compleja mediante un par de polinomios bivariados homogéneos del mismo grado.

Sea $f \in \mathbb{C}(u)$ dada como en (1). El grado de f , al que denotaremos por d , viene dado por $d = \max\{\text{grado}(A), \text{grado}(B)\}$ (teniendo en cuenta que el grado del polinomio nulo se define como $-\infty$). Si $\frac{A}{B}$ es una representación irreducible de f , podemos considerar la extensión de f a $\hat{\mathbb{C}} \cong \mathbf{P}^1(\mathbb{C}) \cong S^2$, \hat{f} , definida como en (2).

En el caso que $\frac{A}{B}$ no sea irreducible, entonces existen valores u_0 (puntos de indeterminación) tales que $A(u_0) = 0 = B(u_0)$. Para solventar este problema, es mejor considerar una nueva ampliación de los modelos señalados con la adición de un punto extra. Es por ello que hemos considerado las extensiones $\mathbf{P}^{1+}(\mathbb{C}) = \mathbf{P}^1(\mathbb{C}) \cup \{\bar{0}\}$ o bien $S^{2+} = S^2 \cup \{(0, 0, 0)\}$.

Con el fin de encontrar una representación de funciones racionales en el modelo $\mathbf{P}^{1+}(\mathbb{C})$ que no presente problemas computacionales, realizamos en (1) el cambio $u = \frac{z}{t}$ ($= [z : t]$). De esta forma, obtenemos la expresión:

$$f\left(\frac{z}{t}\right) = \frac{a_0 + a_1 \frac{z}{t} + \cdots + a_n \left(\frac{z}{t}\right)^n}{b_0 + b_1 \frac{z}{t} + \cdots + b_m \left(\frac{z}{t}\right)^m} = \frac{a_0 t^d + a_1 z t^{d-1} + \cdots + a_n z^n t^{d-n}}{b_0 t^d + b_1 z t^{d-1} + \cdots + b_m z^m t^{d-m}}, \quad (6)$$

donde $d = \max\{n, m\}$. Notemos que ahora tenemos los polinomios bivariados $F(z, t) = a_0 t^d + a_1 z t^{d-1} + \dots + a_n z^n t^{d-n}$ y $G(z, t) = b_0 t^d + b_1 z t^{d-1} + \dots + b_m z^m t^{d-m}$, verificando: $F(\lambda z, \lambda t) = \lambda^d F(z, t)$ y $G(\lambda z, \lambda t) = \lambda^d G(z, t)$, $\lambda \neq 0$.

Definición 2.3.1. Dado un polinomio bivariado $F \in \mathbb{C}[z, t]$ se dice que F es un polinomio bivariado homogéneo si $F = 0$ (el polinomio nulo) o $\exists k \in \mathbb{N}$ tal que se verifica que $F(\alpha z, \alpha t) = \alpha^k F(z, t) \forall \alpha \in \mathbb{C} \setminus \{0\}$ y $\forall z, t \in \mathbb{C}$.

Denotaremos por $\mathbb{C}_h[z, t]$ el subconjunto de los polinomios bivariados homogéneos.

Definición 2.3.2. Dados $F, G \in \mathbb{C}_h[z, t]$, diremos que (F, G) es un par homogéneo si F, G son polinomios bivariados homogéneos satisfaciendo que F y G tienen el mismo grado d . Tal d se llama grado del par homogéneo (F, G) y denotaremos $d = \text{grado}((F, G))$.

Denotaremos por

$$\mathbb{C}_h[z, t] \times \mathbb{C}_h[z, t] = \{(F, G) \in \mathbb{C}_h[z, t] \times \mathbb{C}_h[z, t] \mid (F, G) \text{ es un par homogéneo}\}.$$

Por lo tanto teniendo en cuenta (6), en el modelo $\mathbf{P}^{1+}(\mathbb{C})$ la función racional f se puede representar por la aplicación $[F : G]$, donde $[F : G] : \mathbf{P}^{1+}(\mathbb{C}) \rightarrow \mathbf{P}^{1+}(\mathbb{C})$ viene dada por:

$$[F : G]([z : t]) = [a_0 t^d + a_1 z t^{d-1} + \dots + a_n z^n t^{d-n} : b_0 t^d + b_1 z t^{d-1} + \dots + b_m z^m t^{d-m}]. \quad (7)$$

Es importante observar que con esta expresión f se evalúa cómodamente en los polos, en el punto del infinito ∞ o en los puntos de indeterminación incluso en el caso de que $\frac{A}{B}$ no sea irreducible.

Además, esta representación de una función racional f al considerar numerador y denominador por separado, hace que en el proceso de iteración no se presenten problemas de indefinición o de indeterminación. Esto supone una importante ventaja computacional.

Vamos a describir las funciones que realizan los procesos de representación señalados y que posteriormente hemos implementado en Julia.

En primer lugar, definimos una función cuya misión es la de transformar un polinomio de una variable (u) en un polinomio homogéneo de dos variables (z y t). Si denotamos

$$\mathbb{C}[u] \times \mathbb{N} = \{(p, d) \in \mathbb{C}[u] \times \mathbb{N} \mid \text{grado}(p) \leq d\},$$

podemos definir el siguiente operador de homogenización $H : \mathbb{C}[u] \times \mathbb{N} \rightarrow \mathbb{C}_h[z, t]$:

$$H(p, d) = \begin{cases} a_0 t^d + a_1 z t^{d-1} + \dots + a_n z^n t^{d-n}, & \text{si } p \neq 0, \quad p = a_0 + a_1 z + \dots + a_n z^n \\ 0, & \text{si } p = 0. \end{cases}$$

Para implementar en Julia las funciones racionales, consideramos el isomorfismo vectorial

$$\sum_0^{+\infty} \mathbb{C} \cong \mathbb{C}[u], \quad (a_0, a_1, \dots, a_n) \mapsto a_0 + a_1 u + \dots + a_n u^n.$$

y denotamos

$$\left(\sum_0^{+\infty} \mathbb{C}\right) \times \mathbb{C} \times \mathbb{C}_{\leq \mathbb{N}} = \{((a_0, a_1, \dots, a_n), z, t, d) \mid a_n \neq 0, n \leq d\}.$$

Ahora ya tenemos un contexto adecuado para implementar en Julia las funciones anteriores. En primer lugar, definiremos la función encargada de transformar el polinomio univariado en un polinomio bivariado homogéneo con un grado prefijado:

$$\text{bivariatepolyfunction}: \left(\sum_0^{+\infty} \mathbb{C}\right) \times \mathbb{C} \times \mathbb{C}_{\leq \mathbb{N}} \rightarrow \mathbb{C}$$

dada por:

$$\text{bivariatepolyfunction}((a_0, a_1, \dots, a_n), z, t, d) = a_0 t^d + a_1 z t^{d-1} + \dots + a_n z^n t^{d-n}. \quad (8)$$

La implementación en el lenguaje Julia puede realizarse de varias formas. A continuación, mostramos dos de ellas. La primera está realizada con un bucle *for*, mientras que en la segunda se utiliza un tipo de *for* especial preparado para hacer computación en paralelo.

```
function bivariatepolyfunction{T<:Number,S<:Number,R<:Number}
    (coefficientlist::Array{T,1},
     u::S,
     t::R,
     d::Int)

    ff = coefficientlist[1]*t^d;
    for i = 2:length(coefficientlist)
        ff = ff+coefficientlist[i]*u^(i-1)*t^(d-i+1)
    end
end

@everywhere function bivariatepolyfunctionD{T<:Number,S<:Number,R<:Number}
    (coefficientlist::Array{T,1},
     u::S,
     t::R,
     d::Int)

    return @parallel (+) for i = 1:length(coefficientlist)
        coefficientlist[i]*u^(i-1)*t^(d-i+1)
    end
end
```

Cabe destacar que la segunda manera de resolverlo no es la más eficiente si se realizan pocas iteraciones de la función f . Sin embargo, a medida que el número de iteraciones crece, la velocidad de resolución del bucle es más rápida en función del número de “cores” disponibles. En el caso de usar [JuliaBox \[19\]](#), plataforma que permite ejecutar Julia en cuadernos Jupyter directamente en el navegador, es posible la utilización de 3 “cores” gratuitamente, mientras que si se ejecuta Julia en su línea de comandos dependerá del sistema informático empleado para ello. En ambos casos es necesario ejecutar la orden “addprocs(n-1)” (n representa el número de “cores” disponibles) para poder aprovechar el potencial de todos los “cores”.

Nos centramos ahora en describir la función que hemos implementado en Julia encargada de transformar la función racional en un par de polinomios bivariados,

$$\text{pairoffunctions}: \left(\sum_0^{+\infty} \mathbb{C} \right) \times \left(\sum_0^{+\infty} \mathbb{C} \right) \times \mathbb{C} \times \mathbb{C} \rightarrow \mathbb{C} \times \mathbb{C}$$

dada por:

$$\begin{aligned} \text{pairoffunctions}((a_0, a_1, \dots, a_n), (b_0, b_1, \dots, b_m), z, t) &= \\ &= (\text{bivariatepolyfunction}((a_0, a_1, \dots, a_n), z, t, d), \\ &\quad \text{bivariatepolyfunction}((b_0, b_1, \dots, b_m), z, t, d)) = \\ &= (a_0 t^d + a_1 z t^{d-1} + \dots + a_n z^n t^{d-n}, b_0 t^d + b_1 z t^{d-1} + \dots + b_m z^m t^{d-m}) \\ &\text{(donde } d = \max\{n, m\}). \end{aligned} \tag{9}$$

Dado un par homogéneo $(F, G) \in \mathbb{C}_h[z, t] \times \mathbb{C}_h[z, t]$, podemos considerar las siguientes aplicaciones inducidas:

$$\begin{aligned} (F, G) : \mathbb{C} \times \mathbb{C} &\rightarrow \mathbb{C} \times \mathbb{C}, \text{ dada por: } (F, G)((z, t)) = (F(z, t), G(z, t)) \\ \text{y } [F : G] : \mathbf{P}^1(\mathbb{C}) &\rightarrow \mathbf{P}^1(\mathbb{C}), \text{ dada por: } [F : G]([z : t]) = [F(z, t) : G(z, t)]. \end{aligned}$$

Al calcular las coordenadas homogéneas $(F(z, t), G(z, t))$, pueden aparecer problemas por “underflows”, “overflows” e “indeterminación”. Estas dificultades pueden solventarse utilizando coordenadas homogéneas normalizadas del punto imagen (ver Definición 2.2.1). De manera que podemos definir la siguiente función (donde D representa el 2-disco unidad) que posteriormente hemos implementado en Julia:

$$\text{rationalFunction}: (\mathbb{C}_h[z, t] \times \mathbb{C}_h[z, t]) \times \mathbb{C} \times \mathbb{C} \times \mathbb{N} \rightarrow D \times D$$

dada por:

$$\text{rationalFunction}(F, G, z, t, N) = \text{homogeneousNormalization}((F(z, t), G(z, t)), N). \tag{10}$$

Esta función es la encargada de evaluar un par de polinomios bivariados homogéneos en una pareja de complejos y, posteriormente, obtener las coordenadas homogéneas normalizadas de la imagen para una cierta Tolerancia de Nulidad 10^{-N} prefijada.

2.4. Normalización de un par homogéneo

En esta subsección trataremos la noción de normalización de un par homogéneo (F, G) . Para ello, vamos a introducir algunas nociones y resultados que ayudarán a definirla.

Notemos que si $((a_0, a_1, \dots, a_n), (b_0, b_1, \dots, b_m))$ son los coeficientes de un par homogéneo (F, G) cuyo grado es d y $(z, t) \in D \times D$, se tiene:

$$\begin{aligned} |a_0 t^d + a_1 z t^{d-1} + \dots + a_n z^n t^{d-n}| &\leq |a_0| + |a_1| + \dots + |a_n|, \\ |b_0 t^d + b_1 z t^{d-1} + \dots + b_m z^m t^{d-m}| &\leq |b_0| + |b_1| + \dots + |b_m|. \end{aligned}$$

Lo anterior sugiere las siguientes definiciones.

Definición 2.4.1. Dado $F \in \mathbb{C}_h[z, t]$, donde $F(z, t) = a_0 t^d + a_1 z t^{d-1} + \dots + a_n z^n t^{d-n}$, llamamos norma de F al valor:

$$\|F\| = |a_0| + |a_1| + \dots + |a_n|.$$

Definición 2.4.2. Dado un par homogéneo $(F, G) \in \mathbb{C}_h[z, t] \times \mathbb{C}_h[z, t]$, donde $F(z, t) = a_0 t^d + a_1 z t^{d-1} + \dots + a_n z^n t^{d-n}$ y $G(z, t) = b_0 t^d + b_1 z t^{d-1} + \dots + b_m z^m t^{d-m}$, llamamos norma de (F, G) al valor:

$$\|(F, G)\| = |a_0| + |a_1| + \dots + |a_n| + |b_0| + |b_1| + \dots + |b_m|.$$

Definición 2.4.3. Dado un par homogéneo $(F, G) \in \mathbb{C}_h[z, t] \times \mathbb{C}_h[z, t]$, llamamos normalización de (F, G) y lo denotamos por $(F, G)^*$, al siguiente par:

$$(F, G)^* = \begin{cases} \left(\frac{F}{\|(F, G)\|}, \frac{G}{\|(F, G)\|} \right), & \text{si } (F, G) \neq (0, 0) \\ (0, 0), & \text{si } (F, G) = (0, 0). \end{cases}$$

Diremos que (F, G) está normalizado si $(F, G)^* = (F, G)$.

Notemos que si $(F, G) \in \mathbb{C}_h[z, t] \times \mathbb{C}_h[z, t]$ es un par homogéneo normalizado, entonces para cada $(z, t) \in D \times D$, se tiene que $|F(z, t)| + |G(z, t)| \leq 1$. Por tanto,

$$(F, G)((D \times D)) \subset D \times D.$$

3. Algoritmos utilizados para la representación de cuencas de atracción

A lo largo de las secciones anteriores hemos ido introduciendo algunas técnicas matemáticas y desarrollando algunos aspectos teóricos que utilizaremos para construir programas informáticos con la capacidad de representar gráficamente cuencas de atracción de puntos finales asociados a una función racional determinada, mediante un proceso de iteración.

En la subsección 2.3 hemos visto que un par homogéneo (F, G) representa una función racional $f = [F : G]$ de $\mathbf{P}^{1+}(\mathbb{C})$. En las siguientes subsecciones vamos a mostrar los algoritmos que se han desarrollado para estudiar las cuencas inducidas por f . Algunos de los algoritmos, que mostraremos a continuación, sin programación paralela pueden verse en [6].

3.1. Algoritmo para el cálculo de puntos fijos

Vamos a describir un algoritmo que calculará los puntos fijos de una función racional $f = [F : G]$. Antes, expresamos las nociones de punto fijo, punto fijo propio, punto de indeterminación y punto de indeterminación propio de f en el contexto de $\mathbf{P}^{1+}(\mathbb{C})$.

Definición 3.1.1. Sea $f = [F : G] : \mathbf{P}^{1+}(\mathbb{C}) \rightarrow \mathbf{P}^{1+}(\mathbb{C})$ una aplicación racional y sea $[z_0 : t_0] \in \mathbf{P}^{1+}(\mathbb{C})$. Diremos que $[z_0 : t_0]$ es un punto fijo propio de f si $[z_0 : t_0]$ es un punto fijo de f ($f([z_0 : t_0]) = [z_0 : t_0]$) y $[z_0 : t_0] \neq [0 : 0]$. Diremos que $[z_0 : t_0]$ es un punto de indeterminación de f si $f([z_0 : t_0]) = [0 : 0] = \bar{0}$ y diremos que $[z_0 : t_0]$ es un punto de indeterminación propio de f si $[z_0 : t_0]$ es un punto de indeterminación y $[z_0 : t_0] \neq [0 : 0]$.

Denotaremos los subconjuntos de puntos fijos, puntos fijos propios, puntos de indeterminación y puntos de indeterminación propios, respectivamente, por:

$$\text{Fix}([F : G]) = \{[z_0 : t_0] \mid \exists \lambda \in \mathbb{C} \setminus \{0\}, F(z_0, t_0) = \lambda z_0, G(z_0, t_0) = \lambda t_0\},$$

$$\text{PFix}([F : G]) = \{[z_0 : t_0] \mid (z_0, t_0) \neq (0, 0) \text{ y } [z_0 : t_0] \in \text{Fix}([F : G])\},$$

$$\text{Ind}([F : G]) = \{[z_0 : t_0] \mid F(z_0, t_0) = 0 = G(z_0, t_0)\},$$

$$\text{PInd}([F : G]) = \{[z_0 : t_0] \mid (z_0, t_0) \neq (0, 0) \text{ y } [z_0 : t_0] \in \text{Ind}([F : G])\}.$$

Definición 3.1.2. Sea $(F, G) \in \mathbb{C}_h[z, t] \times \mathbb{C}_h[z, t]$. Diremos que el par (F, G) es irreducible si $(F, G) = (0, 0)$ o verifica que cuando $(H F_1, H G_1) = (F, G)$, entonces $\text{grado}(H) = 0$.

Lema 3.1.1. Sea $f = [F : G] : \mathbf{P}^{1+}(\mathbb{C}) \rightarrow \mathbf{P}^{1+}(\mathbb{C})$. Entonces (F, G) es irreducible si y solo si $\text{Ind}([F : G]) \cap \mathbf{P}^1(\mathbb{C}) = \emptyset$.

Dada una función racional f , vamos a describir el método para obtener sus puntos fijos. Suponemos en lo que sigue que f está representada por un par homogéneo irreducible (F, G) . En función del grado de f , distinguimos los siguientes casos que se pueden presentar: $\text{grado}(f) = -\infty$, $\text{grado}(f) = 0$ y $\text{grado}(f) \geq 1$.

- (a) Si $\text{grado}(f) = -\infty$, entonces $f = [0 : 0]$. En este caso, $\text{Fix}([F : G]) = \{[0 : 0]\}$, $\text{PFix}([F : G]) = \emptyset$, $\text{Ind}([F : G]) = \mathbf{P}^{1+}(\mathbb{C})$ y $\text{PInd}([F : G]) = \mathbf{P}^1(\mathbb{C})$.
- (b) Si $\text{grado}(f) = 0$, entonces $F = a$ y $G = b$ (polinomios constantes) con $(a, b) \neq (0, 0)$. En este caso, tenemos que $\text{Fix}([F : G]) = \{[a : b]\}$ y $\text{PFix}([F : G]) = \{[a : b]\}$.
- (c) Si $\text{grado}(f) \geq 1$:
 - Si $[F : G] = [z : t]$, entonces (F, G) representa la identidad, así que todos los puntos de $\mathbf{P}^{1+}(\mathbb{C})$ son puntos fijos; por tanto, $\text{Fix}([F : G]) = \mathbf{P}^{1+}(\mathbb{C})$ y $\text{PFix}([F : G]) = \mathbf{P}^1(\mathbb{C})$.
 - Si $[F : G] \neq [z : t]$, vamos a analizar si una raíz $(z_0, t_0) \neq (0, 0)$ del polinomio bivariado $zG(z, t) - tF(z, t) = 0$ es un punto fijo de f . Supongamos que $z_0G(z_0, t_0) - t_0F(z_0, t_0) = 0$ y $(z_0, t_0) \neq (0, 0)$. Podemos distinguir:
 - Si $t_0 = 0$ y $z_0 \neq 0$, entonces $z_0G(z_0, t_0) - t_0F(z_0, t_0) = 0 \Leftrightarrow G(z_0, t_0) = 0$; por tanto, $[1 : 0]$ es un punto fijo de f . Como estamos suponiendo que el par (F, G) es irreducible, tenemos que $F(z_0, t_0) \neq 0$.
 - Si $t_0 \neq 0$, podemos tomar $x = z_0/t_0$ y considerar el polinomio $p_1(x) = xG(x, 1) - F(x, 1)$. Más aún, podemos considerar el polinomio mónico $p(x)$ correspondiente al polinomio $p_1(x)$. Notemos que $z_0G(z_0, t_0) - t_0F(t_0, z_0) = 0 \Leftrightarrow p(x) = 0$. Esto es, los puntos fijos de $[F : G]$ se corresponden con las raíces del polinomio p .

Para buscar las raíces de $p(x)$, es posible hacerlo mediante varios métodos. Nosotros consideraremos dos de ellos. Podemos usar el paquete de Julia “SymPy”, que utiliza la función “solve” del paquete “Solvers” de “Python”, o usar el paquete “Polynomials” de Julia. Este último construye la “matriz acompañante” de p a la que denotaremos por $C(p)$. Dicha matriz cuadrada se caracteriza por la propiedad de que las raíces del polinomio p se corresponden con los valores propios de la matriz $C(p)$.

Recordemos que la matriz $C(q)$ correspondiente a un polinomio mónico q :

$$q(x) = q_0 + q_1x + \cdots + q_{n-1}x^{n-1} + x^n,$$

viene dada por:

$$C(q) = \begin{bmatrix} 0 & 0 & \cdots & 0 & -q_0 \\ 1 & 0 & \cdots & 0 & -q_1 \\ 0 & 1 & \cdots & 0 & -q_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -q_{n-1} \end{bmatrix}.$$

Por tanto, si $p^{-1}(0) = \{x_1, x_2, \dots, x_k\}$ son las raíces de p , entonces $\text{PFix}([F : G]) = \{[x : 1] \mid x \in p^{-1}(0)\}$.

Notemos que $G(x, 1) \neq 0$, pues si $G(x, 1) = 0$ entonces $[x : 1]$ sería un punto de indeterminación de $[F : G]$ y esto contradice el hecho de que (F, G) es irreducible.

Utilizando el algoritmo desarrollado previamente desde un punto de vista teórico, se consideran las siguientes aplicaciones que posteriormente implementaremos en Julia.

Dado un conjunto X , denotamos:

$$\text{FiniteArrays}(X) = \bigsqcup_{j=0}^{+\infty} X^j, \text{ donde } X^j = X \times \cdots \times X.$$

Así, podemos definir la aplicación:

$$\text{fixedPointsofaPair}: (\mathbb{C}_h[z, t] \times \mathbb{C}_h[z, t]) \setminus \{[z : t]\} \rightarrow \text{FiniteArrays}(\mathbf{P}^{1+}(\mathbb{C}))$$

dada por:

$$\text{fixedPointsofaPair}((F, G)) = \text{Fix}([F : G]), \quad (11)$$

que, para una función racional f representada por un par homogéneo (F, G) , calcula los puntos fijos de dicha función.

Una implementación en Julia usando computación paralela es la siguiente:

```
@everywhere function fixedPointsofaPairD
    {T<:Number,S<:Number}(numerator::Array{T,1},
                           denominator::Array{S,1})

    numden = samenumberofcoefficients(numerator, denominator);
    num = numden[1];
    den = numden[2];
    l = length(num);
    FG = pairoffunctions(num, den);
    Fix = []; PFix = [];

    if sum(abs, hcat(num, den)) == 0.0
        Fix = union(PFix, [(complex(0.0), complex(0.0))]);
        return Fix, PFix;
    end
end
```

```

elseif l == 1
    Fix = [(num[1],den[1])]; PFix = Fix;
    return Fix, PFix;
elseif l == 2 && abs(num[1]) == 0.0 && abs(den[1]) == 0.0
    && sum(vcat(num[1],-den[1])) == 0.0
    Fix = ["Todos puntos son puntos fijos"];
    PFix = ["Todos puntos son puntos fijos propios excepto (0,0)"];
    return Fix, PFix;
else
    if abs(FG[2](complex(1.0),complex(0.0))) == 0.0
        Fix = push!(PFix,[(complex(1.0),complex(0.0))]);
    end
    x = Sym("x")
    Polys = @parallel (+) for i in 1:l
        [num[i]*x^(i-1), den[i]*x^i]
    end
    roots = solve(Polys[1] - Polys[2],x);
    roots1 = @parallel (hcat) for i in 1:length(roots)
        if (roots[i] != 0)
            vcat([(roots[i],complex(1.0))],
                [(roots[i],complex(1.0))]);
        else
            vcat([(roots[i],complex(1.0))],[]);
        end
    end
    Fixnew = roots1[1,:];
    PFixnew = roots1[2,:];
    while findfirst(PFixnew,()) != 0
        deleteat!(PFixnew,findfirst(PFixnew,()));
    end
    if !isempty(Fix)
        return union(Fix,Fixnew), union(PFix,PFixnew);
    end
    return Fixnew, PFixnew;
end
end
end

```

En el caso que (F, G) no sea irreducible, se puede mejorar el algoritmo anterior para obtener la función:

$$\text{fixedOrindeterminatePointsofaPair}: (\mathbb{C}_h[z, t] \times \mathbb{C}_h[z, t]) \setminus \{[z : t] \cup \{(0, 0)\}\} \rightarrow \text{FiniteArrays}(\mathbf{P}^{1+}(\mathbb{C})),$$

dada por:

$$\text{fixedOrindeterminatePointsofaPair}((F, G)) = \text{fixedPointsofaPair}([F : G]) \sqcup \text{Ind}([F : G]), \quad (12)$$

que, dada una función racional, calcula los puntos fijos y puntos de indeterminación de dicha función.

Una implementación en Julia de la misma es la siguiente:

```
@everywhere function fixedOrindeterminatePointsofaPairD
    {T<:Number,S<:Number}(numerator::Array{T,1},
                           denominator::Array{S,1})

    numden = samenumberofcoefficients(numerator, denominator);
    num = numden[1];
    den = numden[2];
    l = length(num);
    FG = paioffunctions(num, den);
    Fix = []; MFix = []; FixInd = [];

    if sum(abs, hcat(num, den)) == 0.0
        Fix = [(complex(0.0),complex(0.0))];
        FixInd = ["Todos puntos son puntos de indeterminación"];
        return Fix, MFix, FixInd;
    elseif l == 1
        FixInd = [(num[1],den[1])];
        return Fix, MFix, FixInd;
    elseif abs(num[1]) == 0.0 && den[length(den)] == 0.0 &&
        sum(i->abs(num[i] - den[i-1]), 2:l) == 0.0
        Fix = [(complex(0.0),complex(0.0))];
        MFix = [(complex(1.0),complex(0.0))];
        FixInd = ["Todos puntos son puntos fijos"];
        return Fix, MFix, FixInd;
    elseif abs(FG[2](complex(1.0),complex(0.0))) == 0.0 &&
        abs(FG[1](complex(1.0),complex(0.0))) != 0.0
        Fix = [(complex(0.0),complex(0.0))];
        MFix = [(complex(1.0),complex(0.0))];
    elseif abs(FG[2](complex(1.0),complex(0.0))) == 0.0
        && abs(FG[1](complex(1.0),complex(0.0))) == 0.0
        Fix = [(complex(0.0),complex(0.0))];
        FixInd = [(complex(1.0),complex(0.0))];
        end
    root = Polynomials.roots(Polynomials.Poly(num) -
                             Polynomials.Poly(den) *
                             Polynomials.Poly([complex(0.0),
                                                complex(1.0)]));
    root1 = @parallel (vcat) for i in 1:length(root)
        (complex(root[i]),complex(1.0))
    end
    return Fix, MFix, union(FixInd,root1);
end
```

end
end

3.2. Proceso de iteración de una función racional

Dada una función racional $f : \mathbf{P}^{1+}(\mathbb{C}) \rightarrow \mathbf{P}^{1+}(\mathbb{C})$, vamos a describir el proceso de iteración de la misma con el objetivo de obtener finalmente una representación gráfica de la cuenca de atracción de cada punto fijo. Para determinar el punto final asociado a un punto $x = [z : t] \in \mathbf{P}^{1+}(\mathbb{C})$, podemos iterar la función racional f para obtener sucesiones finitas de la forma:

$$(x, f(x), f^2(x), f^3(x), \dots, f^{k-1}(x), f^k(x)).$$

Utilizaremos coordenadas homogéneas normalizadas (z, t) del punto x y un par homogéneo normalizado (F, G) que represente a $f = [F : G]$. Para cierta Tolerancia de Nulidad prefijada, consideramos la función:

$$f_N((z, t)) = \text{homogeneousNormalization}(F(z, t), G(z, t), N).$$

Notemos que si $\lambda \in \mathbb{C}$ verifica que $|\lambda| = 1$, entonces $f_N((z, t)) = f_N(\lambda(z, t))$.

Dado que se está trabajando en un entorno informático, es necesario establecer un número máximo de iteraciones l y también una tolerancia 10^{-C} , $C \in \mathbb{N}$, a la que llamaremos Tolerancia de Cauchy, que determinará cuando una iteración de un punto está suficientemente próxima a la siguiente.

Se dice que un punto $p \in \mathbf{P}^{1+}(\mathbb{C})$ es un punto fijo de f , módulo una Tolerancia de Cauchy 10^{-C} , si se verifica que $d(p, f(p)) < 10^{-C}$. Y un punto $p \in \mathbf{P}^{1+}(\mathbb{C})$ es un punto fijo módulo Tolerancia de Cauchy 10^{-C} y Tolerancia de Nulidad 10^{-N} si se verifica que $d(p, f_N(p)) < 10^{-C}$.

Para generar el proceso de iteración se utiliza el siguiente algoritmo:

Mientras la distancia cordal entre $f_N^k(x)$ y $f_N^{k+1}(x)$ sea mayor que 10^{-C} y el número de iteraciones k sea menor que el número máximo de iteraciones l , entonces vuelve a realizarse una nueva iteración para el punto $f_N^{k+1}(x)$ y se vuelven a comprobar si se cumplen las dos condiciones anteriores.

Por el contrario, si una de las dos condiciones no se cumple, entonces el proceso finaliza devolviendo como resultado $[f_N^{k+1}(x), k]$.

El algoritmo descrito se ha implementado en Julia con la función `newstep`, dada por:

$$\text{newstep}(f, l, N, C, x) = \begin{cases} (f_N^{l+1}(x), l), & \text{si } d(f_N^k(x), f_N^{k+1}(x)) > 10^{-C}, 0 \leq k < l \\ (f_N^{k+1}(x), k), & k = \min\{s \mid d(f_N^s(x), f_N^{s+1}(x)) \leq 10^{-C} \text{ y } s < l\}, \\ & \text{en otro caso.} \end{cases} \quad (13)$$

3.3. Construcción de 2-arrays posición-iteración

Dados un punto p y una lista de puntos $\{p_1, \dots, p_k\}$, la función `positionuptotolerance`, será la encargada de extraer de dicha lista el primer punto que se encuentre a una distancia del punto p menor que una Tolerancia de Aproximación prefijada 10^{-A} , $A \in \mathbb{N}$.

Tal función (que se ha implementado en Julia):

$$\text{positionuptotolerance}: \text{FiniteArrays}(\mathbf{P}^{1+}(\mathbb{C})) \times \mathbb{N} \times \mathbf{P}^{1+}(\mathbb{C}) \rightarrow \mathbb{N}$$

viene dada por:

$$\begin{aligned} \text{positionuptotolerance}((p_1, \dots, p_k), A, p) = \\ = \begin{cases} \min\{r \mid d(p_r, p) < 10^{-A}\}, & \text{si } \{r \mid d(p_r, p) < 10^{-A}\} \neq \emptyset \\ 0, & \text{en otro caso.} \end{cases} \end{aligned} \quad (14)$$

Lo anterior se aplica para definir una función implementada en Julia que para una función racional f y una lista de puntos dada (p_1, \dots, p_k) (que, según los intereses, pueden ser puntos fijos o puntos de indeterminación de f) es la encargada de obtener para cada punto $x \in \mathbf{P}^{1+}(\mathbb{C})$ un par de enteros no negativos (“posición”, “iteración”) donde la “posición” determina el punto de la lista de puntos (p_1, \dots, p_k) que es el límite de la sucesión de iteraciones de la función f en el punto x (en el caso de que converja a un punto fijo). El entero “iteración” representa el número de iteraciones que han sido necesarias para que el punto de la sucesión de iteraciones de f en x se acerque a un punto de la lista $\{p_1, \dots, p_k\}$ para una Tolerancia de Aproximación prefijada 10^{-A} .

Tal función, `positioninterationuptotolerances`, viene dada por:

$$\begin{aligned} \text{positioninterationuptotolerances}(f, (p_1, \dots, p_k), l, N, C, A, x) = \\ = \text{positionuptotolerance}((p_1, \dots, p_k), A, \text{newstep}(f, l, N, C, x)[1], \text{newstep}(f, l, N, C, x)[2]). \end{aligned} \quad (15)$$

3.4. Construcción de entornos rectangulares y representaciones gráficas globales

Nuestro objetivo es poder representar gráficamente cuencas de atracción de puntos finales de una función racional f , considerando entornos rectangulares del origen y del punto ∞ que nos permitirán obtener una visualización global de las cuencas en $\hat{\mathbb{C}}$. Para ello, construiremos un rectángulo dentro del cual crearemos una cuadrícula y a cada vértice de la misma se le aplicará el procedimiento descrito anteriormente para así aproximar las cuencas de atracción de la función f .

Para producir dicha imagen, vamos a usar una resolución del orden de 2^{-k} , $k \in \mathbb{N}$. Además, podemos representar el rectángulo como un array 2-dimensional mediante la función:

$$\text{rectangle}: (\mathbb{R} \times \mathbb{R}) \times (\mathbb{R} \times \mathbb{R}) \times \mathbb{N} \rightarrow \bigsqcup_{p=0, q=0}^{+\infty} ((\mathbf{P}^1(\mathbb{C}))^p)^q$$

dada por:

$$\text{rectangle}((a, b), (c, d), k) = \left(\left[\left(a + \frac{i}{2^k} \right) + \left(c + \frac{j}{2^k} \right) im : 1 \right] \right)_{i,j}, \quad (16)$$

donde im representa el número complejo que verifica que $im^2 = -1$ e i, j son dos números enteros no negativos verificando $\frac{i}{2^k} < |b - a|$ y $\frac{j}{2^k} < |d - c|$, respectivamente.

Podemos implementar dicha función en el lenguaje de programación Julia así:

```
function rectangle(xinterval::Tuple{Float64,Float64} = (-1.5,1.5),
    yinterval::Tuple{Float64,Float64} = (-1.5,1.5),
    expprecision = 10)

    xint = linspace(xinterval[1], xinterval[2],
        floor(-xinterval[1]*2^exprecision +
            xinterval[2]*2^exprecision + 1));
    yint = linspace(yinterval[2], yinterval[1],
        floor(-yinterval[1]*2^exprecision +
            yinterval[2]*2^exprecision + 1));
    return [(complex(r,i),complex(1.0)) for i in yint, r in xint];
end
```

O utilizando programación en paralelo:

```
@everywhere function rectangleD(xinterval::Tuple{Float64,Float64}
    = (-1.5,1.5),
    yinterval::Tuple{Float64,Float64}
    = (-1.5,1.5),
    expprecision = 10)

    xint = linspace(xinterval[1], xinterval[2],
        floor(-xinterval[1]*2^exprecision +
            xinterval[2]*2^exprecision + 1));
    yint = linspace(yinterval[2], yinterval[1],
        floor(-yinterval[1]*2^exprecision +
            yinterval[2]*2^exprecision + 1));
    return @parallel (hcat) for r in xint
        @parallel (vcat) for i in yint
            (complex(r,i),complex(1.0));
        end
    end
end
```

Es importante señalar que las versiones paralelas de las funciones descritas en anteriores subsecciones implementadas en Julia, tienen una particular importancia cuando

se aplican a los vértices de un rectángulo, ya que distribuyen las tareas de cálculo entre los diferentes “cores” que tenga el entorno informático de trabajo utilizado. Por ejemplo, podemos evaluar la función `positioniterationuptotolerances` definida en (15), para cada vértice de un rectángulo construido con (16). La implementación en paralelo viene dada en Julia como sigue:

```
@everywhere function positioniterationuptotolerancesD
    (hpair::Tuple{Function,Function},
    fixedPointList::Array{Tuple{Complex{Float64},
                                Complex{Float64}},1},
    iter::Int64,
    tolerances::Tuple{Int64,Int64,Int64},
    hrectangle::Array{Tuple{Complex{Float64},
                            Complex{Float64}},2})

    size1 = size(hrectangle);
    xint = linspace(1.0, size1[1], size1[1]);
    yint = linspace(1.0, size1[2], size1[2]);
    return @parallel (hcat) for i in xint
        @parallel (vcate) for j in yint
            positioniterationuptotolerances(hpair,
            fixedPointList, iter, tolerances,
            hrectangle[Int(i),Int(j)]);
        end
    end
end
```

En la subsección 1.3.4 de Preliminares, vimos que $\mathbf{P}^1(\mathbb{C}) \cong \hat{\mathbb{C}} \cong S^2$. Una de las propiedades que se deriva de dichos homeomorfismos es que podemos obtener una descripción global de las cuencas de una función racional utilizando dos parametrizaciones rectangulares de la recta proyectiva compleja, que proporcionarán dos imágenes que cubren $\mathbf{P}^1(\mathbb{C})$. Para ello definimos una nueva función:

$$\text{invertedrectangle}: (\mathbb{R} \times \mathbb{R}) \times (\mathbb{R} \times \mathbb{R}) \times \mathbb{N} \rightarrow \bigsqcup_{p=0, q=0}^{+\infty} ((\mathbf{P}^1(\mathbb{C}))^p)^q$$

dada por:

$$\text{invertedrectangle}((a, b), (c, d), k) = \left([1 : (a + \frac{i}{2^k}) + (c + \frac{j}{2^k})im] \right)_{i,j}. \quad (17)$$

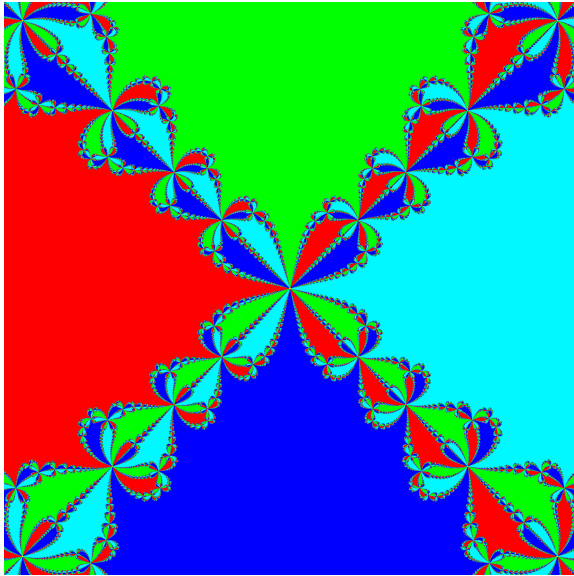
Si tomamos un rectángulo que contenga al disco unidad D , entonces la unión de las imágenes de `rectangle` e `invertedrectangle` “cubren” toda la recta proyectiva compleja. Aún más, usando estas dos expresiones es posible obtener una cuadrícula completa de

$\mathbf{P}^1(\mathbb{C})$; es decir, una estructura de complejo cúbico.

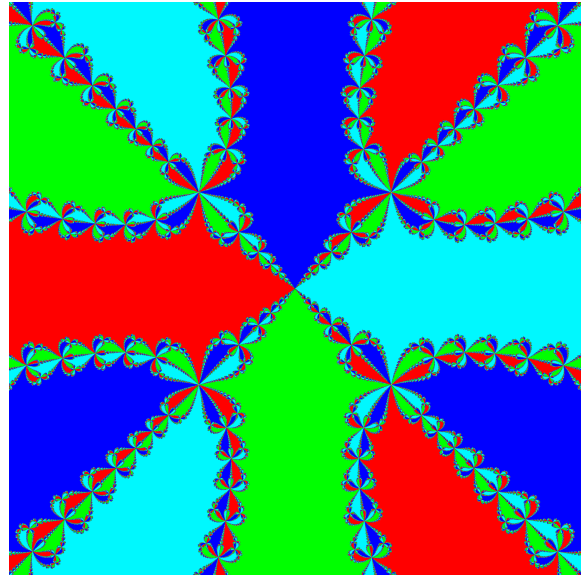
Utilizando los algoritmos anteriores y las funciones implementadas en Julia se obtienen gráficas como las de la Figura 3, que corresponde a la función racional $f : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$ dada por $f(z) = \frac{2z^4+1}{3z^3}$.

Los puntos fijos de f son: ∞ , 1 , -1 , i y $-i$. Además, todos ellos son puntos fijos atractores pues $f^\sharp(\infty) = \frac{2}{3} < 1$ y $f^\sharp(1) = f^\sharp(-1) = f^\sharp(i) = f^\sharp(-i) = \frac{1}{3} < 1$. Los puntos críticos de f son: 1.106682 , -1.106682 , $1.106682i$ y $-1.106682i$.

En la imagen de la izquierda de la Figura 3, se muestra un entorno rectangular del origen $0 \in \hat{\mathbb{C}}$ y en la de la derecha un entorno rectangular del $\infty \in \hat{\mathbb{C}}$. En ambos rectángulos se representan con diferentes colores las distintas cuencas de puntos finales, cuya unión es el conjunto de Fatou y su frontera común es el conjunto de Julia.



Cuencas en un entorno de 0.



Cuencas en un entorno de ∞ .

Figura 3: Cuencas de la función racional $f(z) = \frac{2z^4+1}{3z^3}$.

4. Ejemplos de Conjuntos de Julia

En esta sección comenzaremos introduciendo ciertos conceptos y resultados que utilizaremos a la hora de analizar algunos ejemplos de conjuntos de Julia construidos mediante algoritmos computacionales implementados con el software informático Julia descritos a lo largo de las secciones anteriores. Estos ejemplos corresponden a distintos tipos de funciones polinómicas cuadráticas y cúbicas que aparecen en el artículo “Dynamics in one complex variable” de John Milnor [10]. Para más información de éstos y otros ejemplos puede verse [11]. Además puede verse en [1] la teoría básica de iteración de funciones racionales y en [12] un estudio profundo de las funciones cuadráticas.

John Willard Milnor (1931-) es un matemático estadounidense conocido por sus trabajos en topología diferencial y sistemas dinámicos, entre otros. Es un ilustre investigador, profesor y co-director del Instituto de Ciencias Matemáticas de la Universidad del Estado de Nueva York en Stony Brook. Es uno de los cuatro matemáticos que han ganado la Medalla Fields (1962), el Premio Wolf (1989) y el Premio Abel (2011).



Figura 4: John Milnor.

Señalemos que en esta sección, si bien los algoritmos utilizados para realizar las representaciones gráficas de conjuntos de Julia se enmarcan, para evitar problemas computacionales, en los modelos $\mathbf{P}^{1+}(\mathbb{C}) \cong S^{2+}$ (como hemos comentado en las secciones anteriores), en los ejemplos que mostraremos a continuación los datos que se ofrecerán van a venir expresados en el modelo más familiar $\hat{\mathbb{C}}$. Por ejemplo, hablaremos de ∞ en lugar de $[1 : 0]$ o de 0 en vez de $[0 : 1]$.

Además, como en todos los casos el complementario de la cuenca de ∞ es un conjunto acotado, se pueden dar representaciones globales del conjunto de Julia utilizando únicamente un rectángulo.

4.1. Funciones racionales hiperbólicas y expansivas

Introduciremos en esta subsección una serie de resultados que nos permitirán diferenciar el tipo de funciones descritas en cada uno de los ejemplos. Comenzamos incluyendo varias definiciones relacionadas con funciones expansivas y contractivas.

Definición 4.1.1. Sean $f : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$ una función racional y $X \subset \hat{\mathbb{C}}$ un subconjunto. Diremos que X es f -invariante si $f^{-1}(X) \subset X$.

Definición 4.1.2. Sean $f : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$ una función racional y $X \subset \hat{\mathbb{C}}$ un subconjunto compacto f -invariante. Diremos que f es expansiva en X si existen números reales positivos c y λ , con $\lambda > 1$ tales que

$$(f^n)^\sharp(z) \geq c\lambda^n, \quad n \geq 1,$$

para todo $z \in X$.

Diremos que f es contractiva en X si existen números reales positivos d y μ , con $\mu < 1$ tales que

$$(f^n)^\sharp(z) \leq d\mu^n, \quad n \geq 1,$$

para todo $z \in X$.

Definición 4.1.3. Sea $f : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$ una función racional y sea $X \subset \hat{\mathbb{C}}$ un subconjunto compacto f -invariante. Diremos que f es hiperbólica en X si f es expansiva o contractiva en X , o si X es la unión disjunta de un subconjunto compacto f -invariante, X^+ , en el cual f es expansiva y un subconjunto compacto f -invariante, X^- , en el que f es contractiva.

Puesto que para una función racional $f : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$, el conjunto de Julia $J(f)$ es compacto y f -invariante, si f es expansiva en $J(f)$, entonces f es hiperbólica en $J(f)$.

En los siguientes ejemplos vamos a utilizar la siguiente caracterización de las funciones expansivas (ver [10]):

Teorema 4.1.1. Dada una aplicación racional $f : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$ de grado $d \geq 2$, las siguientes dos condiciones son equivalentes:

- i. f es expansiva en el conjunto de Julia $J(f)$.
- ii. La órbita de cada punto crítico de f converge hacia un ciclo atractor.

Dado que la mayor parte de las funciones racionales que analizaremos en los ejemplos que mostraremos a continuación son de la forma $f(z) = z^2 + c$, $c \in \mathbb{C}$, notemos que se tiene la siguiente propiedad:

Proposición 4.1.1. Una función racional $f : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$ definida por $f(z) = z^2 + c$, $c \in \mathbb{C}$, no puede poseer en \mathbb{C} dos puntos fijos atractores. Además, si un punto fijo de \mathbb{C} es atractor, entonces el otro punto fijo de \mathbb{C} es repulsor.

4.2. Conjunto de Julia diferencialmente parametrizable

Comenzaremos con un ejemplo bastante sencillo. Sea $f : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$ la aplicación dada por $f(z) = z^2$.

Los puntos fijos de f son $\{0, 1, \infty\}$. Además, puesto que la derivada y la derivada esférica de f vienen dadas por $f'(z) = 2z$ y $f^\sharp(z) = 2|z| \frac{1+|z|^2}{1+|z^2|^2}$, se tiene que los únicos puntos críticos son 0 y ∞ , y tanto 0 como ∞ son puntos fijos super-atractores ($f^\sharp(0) = 0 = f^\sharp(\infty)$), mientras que 1 es un punto fijo repulsor ($f^\sharp(1) = 2 > 1$).

Dado un punto z_0 , su órbita $(z_0, f(z_0), f(f(z_0)), \dots)$ tiene el siguiente comportamiento, dependiendo de la relación de z_0 con la circunferencia unidad S^1 :

- (a) Si $|z_0| < 1$, la órbita converge a 0.
- (b) Si $|z_0| = 1$, toda la órbita está contenida en S^1 .
- (c) Si $|z_0| > 1$, la órbita converge a ∞ .

Por tanto, la 2-bola unidad B está contenida en el conjunto de Fatou de f . De manera similar, $\hat{\mathbb{C}} \setminus D$, donde D es el disco unidad, está contenido en el conjunto de Fatou.

Por otro lado, si z_0 pertenece a S^1 , entonces, en cualquier entorno de z_0 existen puntos de B y de $\hat{\mathbb{C}} \setminus D$ y esto implica que $z_0 \in J(f)$. Así que en este caso, el conjunto de Fatou es $F(f) = \hat{\mathbb{C}} \setminus S^1$ y el conjunto de Julia es $J(f) = S^1$. Por lo tanto se puede dar una parametrización al conjunto de Julia S^1 de modo que su inclusión en \mathbb{C} sea una aplicación diferenciable. Así que en este ejemplo, tenemos que el conjunto de Julia es una subvariedad diferenciable de \mathbb{C} .

Notemos que como los únicos puntos críticos de f son 0 e ∞ , que son super-atractores, se tiene, aplicando el Teorema 4.1.1, que f es expansiva en el conjunto de Julia.

Mediante el programa implementado en Julia obtenemos la representación gráfica de las cuencas en la Figura 5. Las zonas amarilla y verde constituyen el conjunto de Fatou (la zona amarilla es la cuenca del ∞ y la verde es la del 0). Su frontera común es la circunferencia S^1 que corresponde el conjunto de Julia.

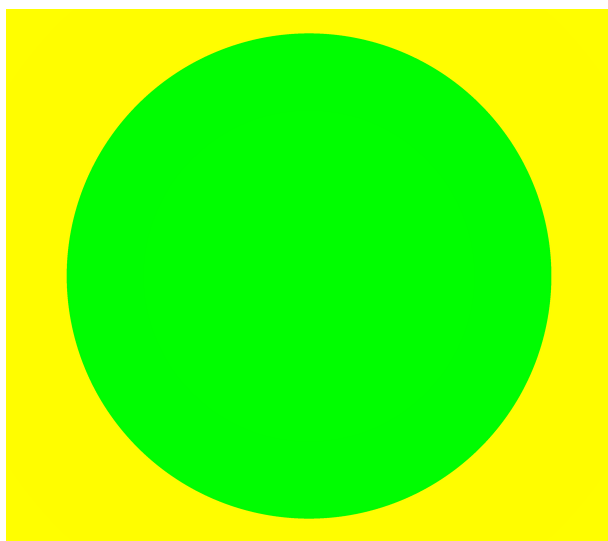


Figura 5: Conjunto de Julia para $z \mapsto z^2$.

Se omite en este caso (como en otros ejemplos posteriores) el entorno rectangular de ∞ (gráficamente estará completamente pintado de amarillo) que junto con el mostrado en la Figura 5 proporciona el cubrimiento completo de $\hat{\mathbb{C}}$.

4.3. Conjunto de Julia topológicamente parametrizable

A continuación, vamos a describir el conjunto de Julia para la aplicación racional $f : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$ dada por $f(z) = z^2 + (0.99 + 0.14i)z$. Este es un ejemplo de lo que Milnor

denomina curva de Jordan con forma “salvaje”.

Recordamos la noción de curva de Jordan y un resultado interesante sobre este tipo de curvas:

Definición 4.3.1. *Una curva de Jordan es una curva plana cuya imagen es homeomorfa a S^1 .*

Si γ es una curva de Jordan, muchas veces se le llama también así a su imagen $\Gamma = \text{Im}(\gamma)$.

Teorema 4.3.1. *(Teorema de Jordan-Schönflies) Toda curva de Jordan Γ divide a \mathbb{C} en dos componentes conexas Ω y Ω' , cuya frontera común es la curva Γ . Una de ellas Ω es acotada y la otra, Ω' es no acotada. Además, $\Omega \cup \Gamma \cong D$.*

El Teorema de la curva de Jordan (resultado anterior suprimiendo la última expresión) lleva el nombre en honor al matemático francés Camille Jordan (1838-1922), quien dió la primera demostración del mismo en 1887. La validez de su demostración fue cuestionada por sus contemporáneos y consideraron que la primera prueba rigurosa del resultado fue dada por el matemático estadounidense Oswald Veblen (1880-1960) en 1905 [16]. Una forma más fuerte del teorema, que es la que nosotros recogemos, fue dada por el matemático alemán Arthur Moritz Schönflies (1853-1928) en 1906. Su prueba contenía un pequeño error que fue corregido por el matemático holandés L. E. J. Brouwer (1881-1966) en 1909 (para más información de ambos teoremas y sus demostraciones, puede verse [15]).

Si analizamos la función racional f con los algoritmos desarrollados, obtenemos la gráfica de la Figura 6, dibujada en la región $(-1.5, 0.6) \times (-1.2, 1.1)$.

En este caso, hay tres puntos fijos $\{0, \infty, 0.01 - 0.14i\}$, de los cuales, 0 es atractor ($f^\#(0) = 0.999985 < 1$), $0.01 - 0.14i$ es repulsor ($f^\#(0.01 - 0.14i) = 1.01966 > 1$) e ∞ es superatractor ($f^\#(\infty) = 0$). Además de ser ∞ un punto crítico, el otro punto crítico de f es $-0.495 - 0.07i$ (que no está en la cuenca del ∞) cuyo valor crítico es $-0.240125 - 0.0693i$.

En nuestro programa, el color negro está reservado para los puntos que (computacionalmente) no convergen a un punto fijo, y el color amarillo se ha reservado para la cuenca del ∞ ; por tanto, el conjunto de Julia $J(f)$ es la frontera de la región amarilla. Dado que el punto fijo 0 es “casi indiferente”, la órbita de un punto muy próximo a cero converge (siguiendo una espiral) muy lentamente a 0. Este hecho sugiere que la zona negra sea la cuenca de atracción de 0, pero las órbitas convergen tan lentamente a 0, debido al valor de $f^\#(0)$, que el ordenador no es capaz de detectar la convergencia en el número máximo de iteraciones prefijadas.

Según Milnor, en este caso existe un homeomorfismo entre S^1 y el conjunto de Julia $J(f)$. Por tanto, éste es una curva de Jordan y tiene las mismas propiedades topológicas que S^1 ($J(f)$ es compacto y arcoconexo). Por tanto, aplicando el Teorema 4.3.1, $J(f)$ divide al plano en dos componentes, una acotada (zona negra) y otra no acotada (zona amarilla), cuya unión es el conjunto de Fatou $F(f)$. Aún más, puede asegurarse que la zona negra es topológicamente equivalente a un disco.

En este caso, el conjunto de Julia $J(f)$ es una subvariedad topológica de \mathbb{C} , aunque según afirma Milnor no es una subvariedad diferenciable (debido a la forma “salvaje” que presenta $J(f)$ a pesar de ser topológicamente equivalente a una circunferencia).

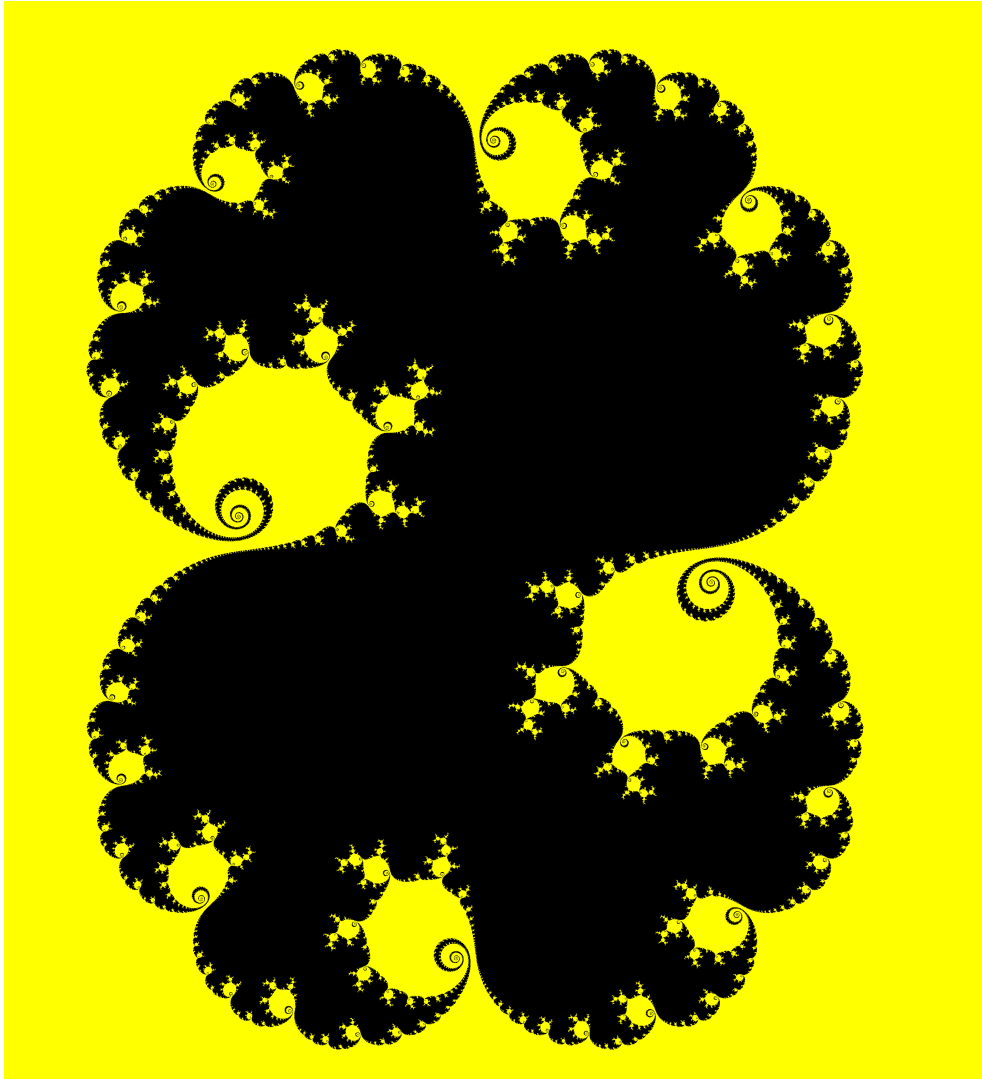


Figura 6: Conjunto de Julia para $z \mapsto z^2 + (0.99 + 0.14i)z$.

4.4. Conjunto de Julia de una función racional no expansiva

En la Figura 7, se describe el conjunto de Julia para la aplicación $f : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$ dada por $f(z) = z^2 + i$. Ésta muestra un ejemplo de lo que Milnor denomina “dentrita”.

La lista de puntos fijos está formada por $\{\infty, 1.30024 - 0.624811i, -0.300243 + 0.624811i\}$, de los cuales ∞ es super-atractor ($f^\#(\infty) = 0$) y los dos restantes son repulsores ya que $f^\#(1.30024 - 0.624811i) = 2.88515 > 1$ y $f^\#(-0.300243 + 0.624811i) = 1.38641 > 1$.

Por otro lado, los puntos críticos de f son ∞ y 0 , cuyos valores críticos son ∞ e i ,

respectivamente. La órbita del punto crítico ∞ es obviamente (∞) ya que es un punto fijo, y podemos calcular la órbita del otro punto crítico 0: $(0, i, -1 + i, -i, -1 + i, -i, \dots)$. Vemos que la órbita de 0 es finita y, por tanto, converge al 2-ciclo $(-1 + i, -i)$, que es repulsor pues $f^\sharp(-1 + i) \cdot f^\sharp(-i) = 4\sqrt{2} > 1$. De esta forma, utilizando el Teorema 4.1.1, podemos afirmar que la función f es no expansiva en el conjunto de Julia $J(f)$.

Por último, notemos que la Figura 7 parece indicar que tanto el conjunto de Julia $J(f)$ (zona roja) como el conjunto de Fatou $F(f)$ (zona amarilla, correspondiente con la cuenca del ∞), dibujados en la región $(-1.5, 1.5) \times (-1.25, 1.25)$, son conjuntos conexos.

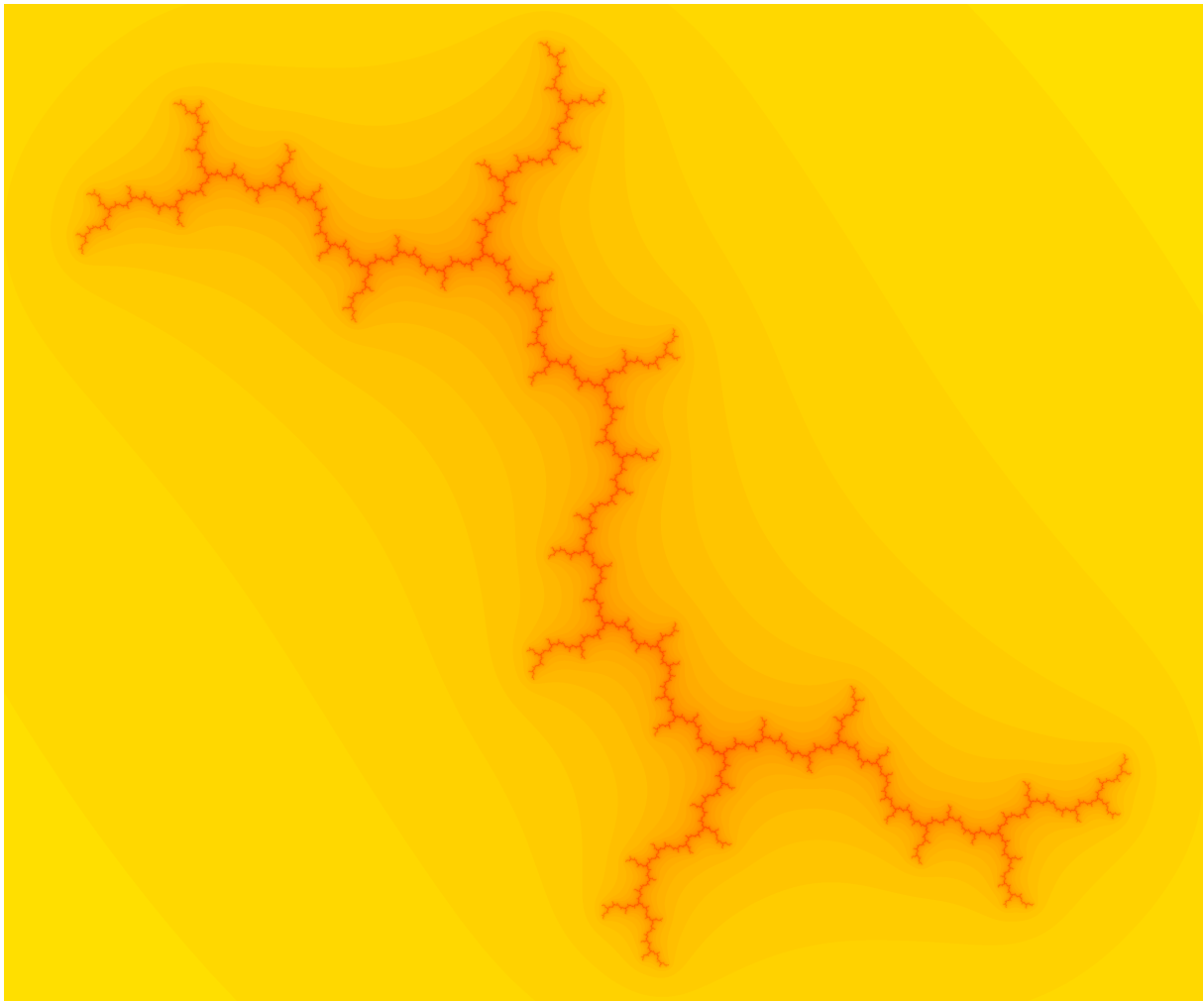


Figura 7: Una “dentrita”, conjunto de Julia para $z \mapsto z^2 + i$.

4.5. Conjunto de Julia de tipo Cantoriano

En este ejemplo vamos a describir el conjunto de Julia para la aplicación $f : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$ dada por $f(z) = z^2 - 0.765 + 0.12i$. Éste es un ejemplo de lo que Milnor denomina conjunto de Cantor “de tipo grueso”.

Analizando la función racional f con los algoritmos desarrollados, obtenemos la gráfica de la Figura 8 dibujada en la región $(-1.55, 1.55) \times (-0.9, 0.9)$. En este caso, los puntos fijos de f son: ∞ , $1.50922 - 0.0594516i$ y $-0.509225 + 0.0594516i$. El punto ∞ es super-atractor ($f^\#(\infty) = 0$) y los otros dos puntos fijos son repulsores ($f^\#(1.50922 - 0.0594516i) = 3.02079 > 1$ y $f^\#(-0.509225 + 0.0594516i) = 1.025367 > 1$).

Los puntos críticos son ∞ y 0 . La órbita de ∞ es obviamente (∞) y la órbita de 0 viene dada por $(0, -0.765 + 0.12i, -0.194175 - 0.0636i, \dots, 29.396587 - 4.73823i, 840.9435 - 278.4553i, 629647.9051 - 468330.25896i, \dots)$ (son necesarias 29 iteraciones para obtener el último valor mostrado en la órbita). Vemos que esta órbita converge al punto fijo super-atractor ∞ y, por lo tanto, aplicando el Teorema 4.1.1, tenemos que la función f es expansiva en el conjunto de Julia $J(f)$.

El conjunto de Julia está formado por la zona de color rojo, luego el conjunto de Fatou viene dado por la región pintada de amarillo. Una de las propiedades que más puede llamar la atención y que remarca Milnor (y parece ser la característica por la que da dicho nombre a la figura) es que existe un homeomorfismo entre el Conjunto de Cantor y $J(f)$. Por lo tanto, $J(f)$ es totalmente desconexo (del cardinal del continuo). Además, podemos observar que $F(f)$ consiste en “arrancar” a una esfera un conjunto de Cantor y por lo tanto es una 2-variedad conexa no compacta.

En comparación con la Figura 7, la estructura topología del conjunto de Julia ha cambiado radicalmente, pues éste antes era conexo y ahora es totalmente desconexo y posee un número no contable de componentes.

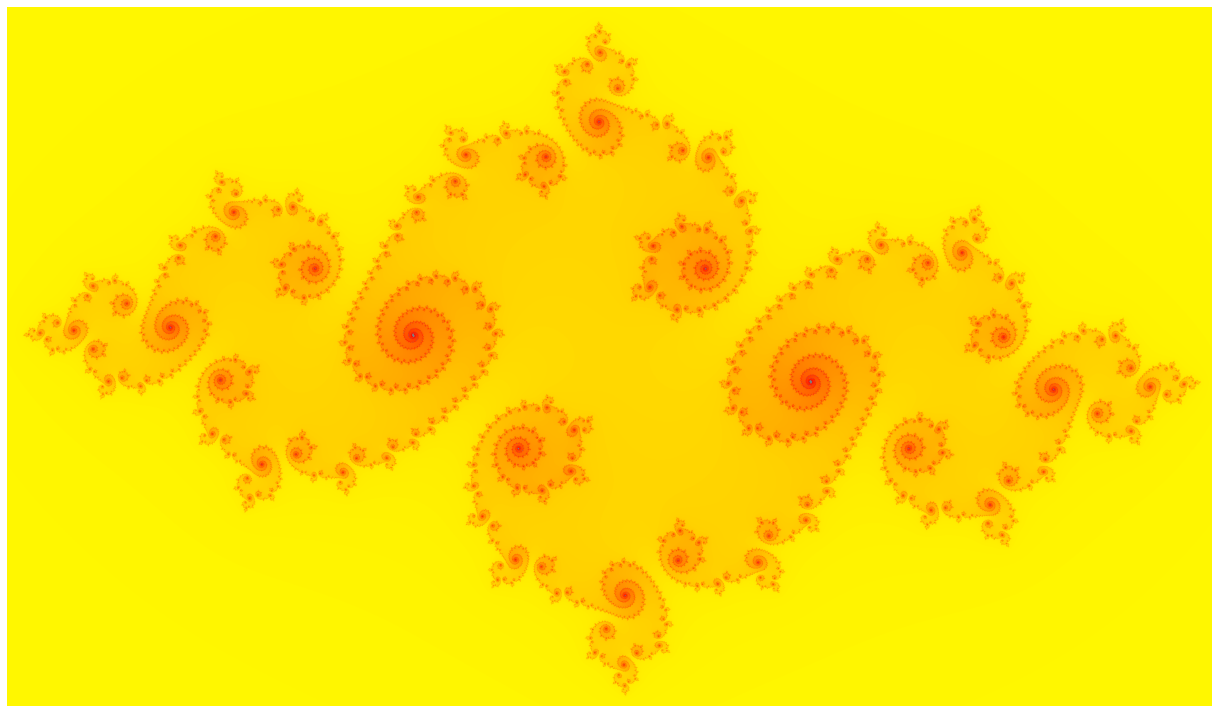


Figura 8: Conjunto de Julia totalmente desconexo para $z \mapsto z^2 - 0.765 + 0.12i$.

4.6. Conjunto de Julia de una función racional expansiva

En el siguiente ejemplo se muestra el conjunto de Julia para la aplicación $f : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$ dada por $f(z) = z^2 - 1.75488$. La imagen de dicho conjunto puede verse en la Figura 9 y es un ejemplo de lo que Milnor denomina “avión”.

La función racional f , la cuál produce la imagen dibujada en la región $(-2, 2) \times (-0.6, 0.6)$, tiene como punto fijos: $\infty, 1.915939$ y -0.915938 . De éstos, ∞ es super-atractor ($f^\#(\infty) = 0$) y tanto 1.915939 como -0.915938 son repulsores, puesto que $f^\#(1.915939) = 3.831861 > 1$ y $f^\#(-0.915938) = 1.831877 > 1$. Además, los puntos críticos de f son ∞ y 0 , cuyo valores críticos son ∞ y -1.75488 , respectivamente.

La órbita de ∞ es (∞) por ser punto fijo y la órbita del punto crítico 0 viene dada por: $(0, -1.75488, 1.324724, 0, -1.75488, 1.324724, \dots)$. Vemos que se trata del 3-ciclo $(0, -1.75488, 1.324724)$, que es super-atractor, ya que $f^\#(0) \cdot f^\#(-1.75488) \cdot f^\#(1.324724) = 0$. Por tanto, aplicando el Teorema 4.1.1, la función f es expansiva en el conjunto de Julia.

El conjunto de Julia está formado por la frontera de la zona amarilla (que corresponde a la cuenca del ∞) y, por lo tanto, el conjunto de Fatou viene dado por la unión del interior de las regiones pintadas de amarillo y de negro. La representación gráfica parece indicar que el conjunto de Julia $J(f)$ es conexo y el de Fatou $F(f)$ tiene infinitas componentes conexas (la amarilla y cada una de las zonas negras).

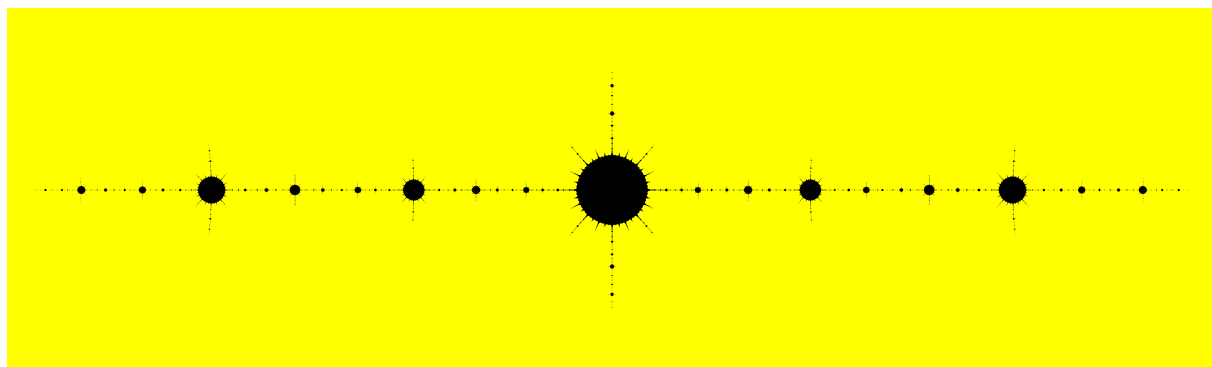


Figura 9: Conjunto de Julia para $z \mapsto z^2 - 1.75488$, el “avión”.

4.7. Conjunto de Julia de un polinomio cúbico

Vamos a describir el conjunto de Julia para la aplicación $f : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$ dada por $f(z) = z^3 + \frac{12}{25}z + \frac{116}{125}i$. Éste es, de todos los ejemplos propuestos por Milnor en [10], el único correspondiente a un polinomio cúbico.

Analizando la función racional f con los algoritmos desarrollados, obtenemos la gráfica de la Figura 10 dibujada en la región $(-1.15, 1.15) \times (-1.15, 1.5)$.

En este caso, hay cuatro puntos fijos: ∞ , $1 - 0.4i$, $-1 - 0.4i$ y $0.8i$; de los cuales ∞ es super-atractor ($f^\sharp(\infty) = 0$) y los otros tres son repulsores: $f^\sharp(1 - 0.4i) = 3.841875 > 1$, $f^\sharp(-1 - 0.4i) = 3.841875 > 1$ y $f^\sharp(0.8i) = 1.44 > 1$.

Por lo que respecta a los puntos críticos, f posee tres: ∞ , $0.4i$ y $-0.4i$.

En el caso de ∞ , sabemos que su órbita es (∞) .

Para el punto crítico $0.4i$, su órbita es $(0.4i, 1.056i, 0.2573i, 1.0345i, 0.3175i, 1.0484i, 0.2789i, 1.0402i, \dots, 0.29278i, 1.0434395i, 0.29279i, 1.0434394i, \dots)$. Luego converge a un 2-ciclo atractor ($f^\sharp(0.29279i) \cdot f^\sharp(1.0434394i) = 0.620923 < 1$), pero con convergencia lenta, pues necesitamos llegar a la iteración 55 para que la distancia entre iteraciones sucesivas sea menor que 10^{-6} .

Para el punto crítico $-0.4i$, obtenemos la órbita $(-0.4i, 0.8i, 0.8i, \dots)$. Es decir, converge al punto fijo repulsor $0.8i$ y con convergencia muchísimo más rápida que en el caso anterior, pues en la segunda iteración ya hemos llegado al punto fijo $0.8i$.

Dado que el último punto crítico de f analizado converge a un punto fijo repulsor, aplicando el Teorema 4.1.1, podemos afirmar que f es una función no expansiva en $J(f)$.

El conjunto de Julia está formado por la frontera de la zona amarilla (que corresponde a la cuenca del ∞) que coincide con la frontera de la zona pintada de negro. El conjunto de Fatou, viene dado por la unión de los interiores de la región pintada de amarillo junto con la zona coloreada de negro.

La Figura 10 parece indicar que el conjunto de Julia $J(f)$ es conexo y además, el conjunto de Fatou $F(f)$ posee infinitas componentes conexas (pero en una cantidad numerable), cada una de ellas homeomorfa a una bola.

Una de las características que llama la atención al observar la gráfica es que, en todos los lugares donde hay ramificaciones, se producen cuatro bifurcaciones, excepto en el que corresponde al punto $0.8i$ donde sólo hay dos bifurcaciones.

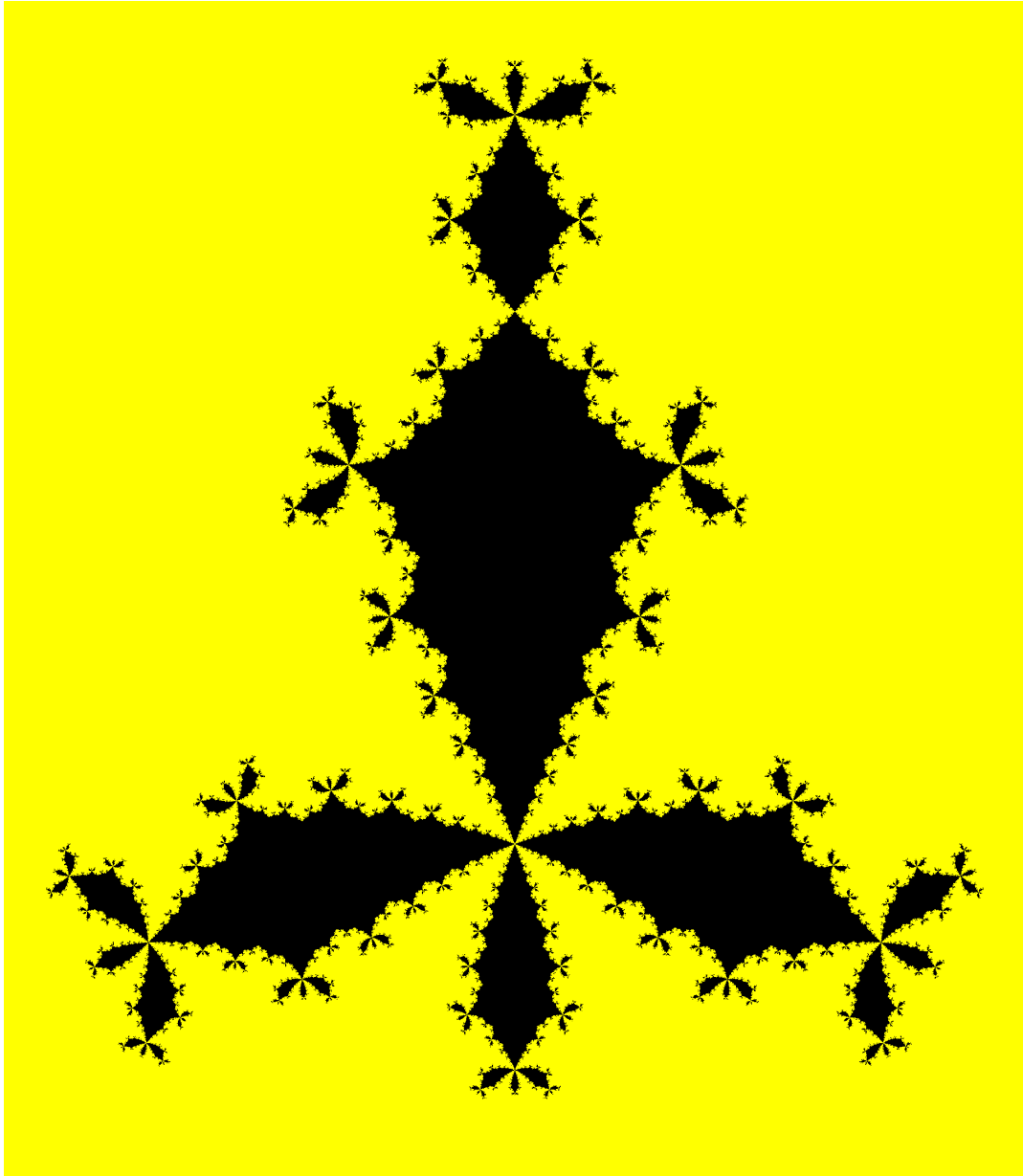


Figura 10: Conjunto de Julia para $z \mapsto z^3 + \frac{12}{25}z + \frac{116}{125}i$.

4.8. Conjunto de Julia de tipo espiral logarítmica

En la Figura 11 aparece el conjunto de Julia para la aplicación $f : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$ dada por $f(z) = z^2 - 0.744336 + 0.121198i$. En este ejemplo se aprecia la estructura de tipo espiral logarítmica que comenta Milnor en su artículo.

Los puntos fijos de la función f son: ∞ , $-0.4990073 + 0.0606592i$ y $1.499003 - 0.0606592i$. De éstos, ∞ es super-atractor ($f^\sharp(\infty) = 0$) y tanto $-0.4990073 + 0.0606592i$ como $1.499003 - 0.0606592i$ son repulsores, pues $f^\sharp(-0.4990073 + 0.0606592i) = 1.005361 > 1$ y $f^\sharp(1.499003 - 0.0606592i) = 3.000468 > 1$.

Además de ser ∞ un punto crítico, el otro punto crítico de f es 0. La órbita de ∞ es (∞) y la órbita de 0 es $(0, -0.744336 + 0.121198i, -0.20499 - 0.05923i, -0.70582 + 0.14548i, -0.267314 - 0.084167i, \dots, -0.7405012 + 0.252003i, -0.2594996 - 0.2520193i, -0.7405097 + 0.251996i, -0.259483 - 0.252013i, \dots)$, que converge a un 2-ciclo repulsor, pues $f^\#(-0.7405097 + 0.251996i) \cdot f^\#(-0.259483 - 0.252013i) = 1.13177 > 1$, pero con una convergencia muy lenta. En efecto, los últimos valores mostrados de la órbita han sido obtenidos con 55 iteraciones y siguen observándose unas distancias entre ellos del orden de 10^{-5} . Además, como el último punto crítico converge hacia un 2-ciclo repulsor, aplicando el Teorema 4.1.1, podemos afirmar que f es una función no expansiva en el conjunto de Julia.

En la Figura 11 se muestra el conjunto de Julia, dibujado en la región $(-1.5, 1.5) \times (-1, 1)$, formado por la frontera de la zona amarilla (correspondiente a la cuenca de ∞). Además, el conjunto de Fatou viene dado por la unión de los interiores de la región pintada de amarillo y la pintada de negro, color reservado para aquellos puntos que no convergen computacionalmente.

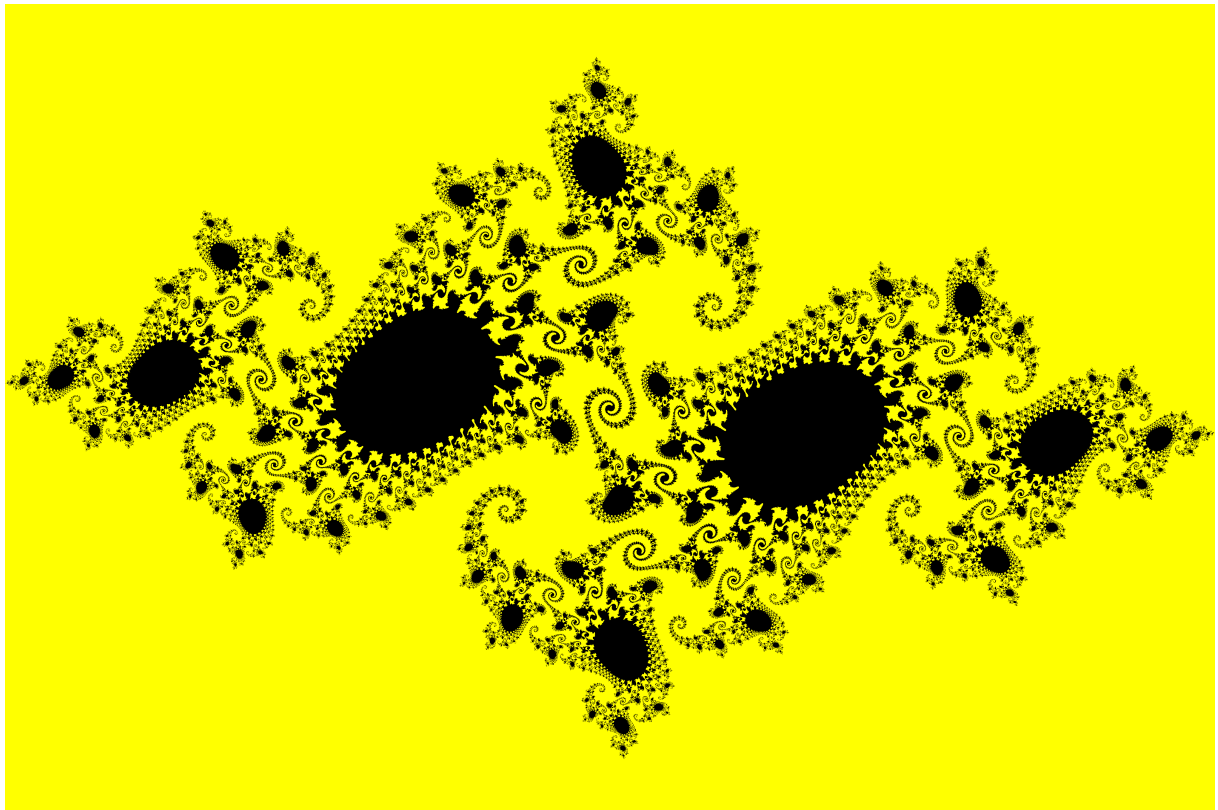


Figura 11: Conjunto de Julia para $z \mapsto z^2 - 0.744336 + 0.121198i$.

En la Figura 12 se muestra un zoom del conjunto de Julia de la función f , dibujado en la región $(0.185, 0.385) \times (0.175, 0.34)$, donde se aprecia su estructura espiral.

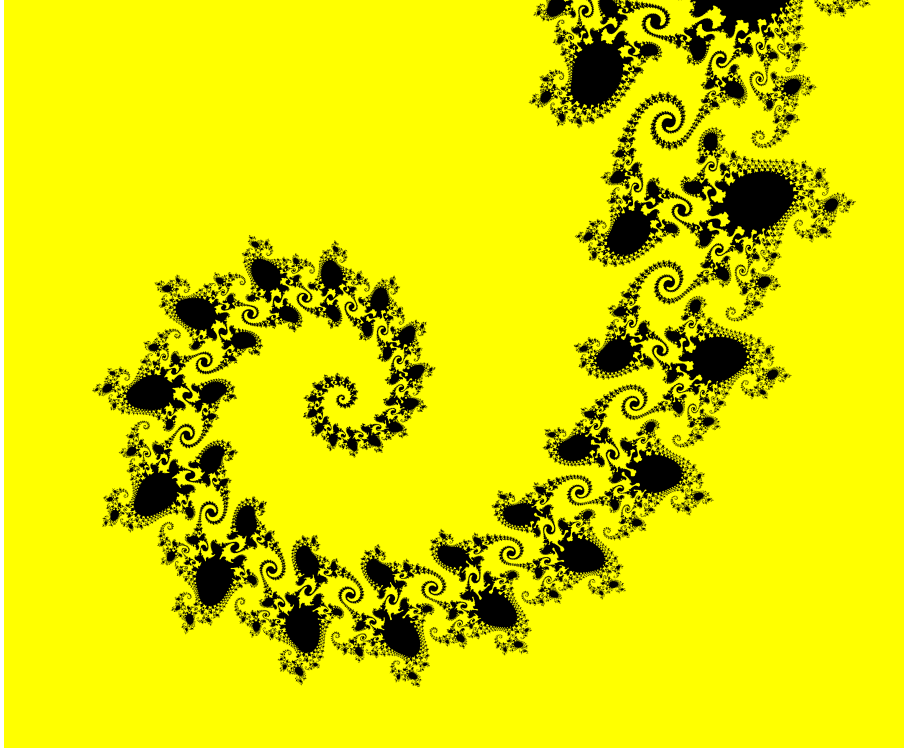


Figura 12: Detalle del conjunto de Julia para $z \mapsto z^2 - 0.744336 + 0.121198i$.

4.9. Conjunto de Julia de tipo arbóreo

Describimos a continuación el conjunto de Julia para la aplicación $f : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$ dada por $f(z) = z^2 + 0.424513 + 0.207530i$. Éste es un ejemplo, junto con el de la Figura 11, de lo que Milnor denomina estructura de tipo espiral logarítmica.

Si analizamos la función racional f con los algoritmos anteriormente descritos, obtenemos la gráfica de la Figura 13, dibujada en la región $(-1.25, 1.25) \times (-1.1, 1.1)$.

En este caso, hay tres puntos fijos: ∞ , $0.280183 + 0.472051i$ y $0.719817 - 0.472051i$. De los cuales, los dos últimos son repulsores, pues $f^\#(0.280183 + 0.472051i) = 1.0978798 > 1$ y $f^\#(0.719817 - 0.472051i) = 1.721591 > 1$ e ∞ es super-atractor ya que $f^\#(\infty) = 0$.

En cuanto a los puntos críticos de f , además del ∞ , 0 es el otro punto crítico. En el caso de ∞ , su órbita es (∞) , y la órbita de 0 es: $(0, 0.424513 + 0.207530i, 0.56166 + 0.3837284i, 0.592723 + 0.6385764i, 0.368053 + 0.96453i, -0.3703365 + 0.917525i, -0.2801895 - 0.4720559i, 0.280182 + 0.4720601i, 0.280174 + 0.47206i, \dots, 0.280154 + 0.471947i, 0.280265 + 0.471965i, \dots)$, que converge al punto fijo repulsor $0.280183 + 0.472051i$, con una convergencia muy lenta (las últimas iteraciones mostradas corresponden a la iteración 55 con una distancia entre ellas del orden de 10^{-3}). Así, aplicando el Teorema 4.1.1, tenemos que f es una función no expansiva en el conjunto de Julia.

El conjunto de Julia está formado por la zona pintada de rojo que es la frontera de la zona amarilla (que corresponde a la cuenca del ∞)y, por tanto, el conjunto de Fatou viene dado por la región pintada de amarillo. La Figura 13 parece indicar que tanto el conjunto de Julia $J(f)$ como el de Fatou $F(f)$ son conexos.

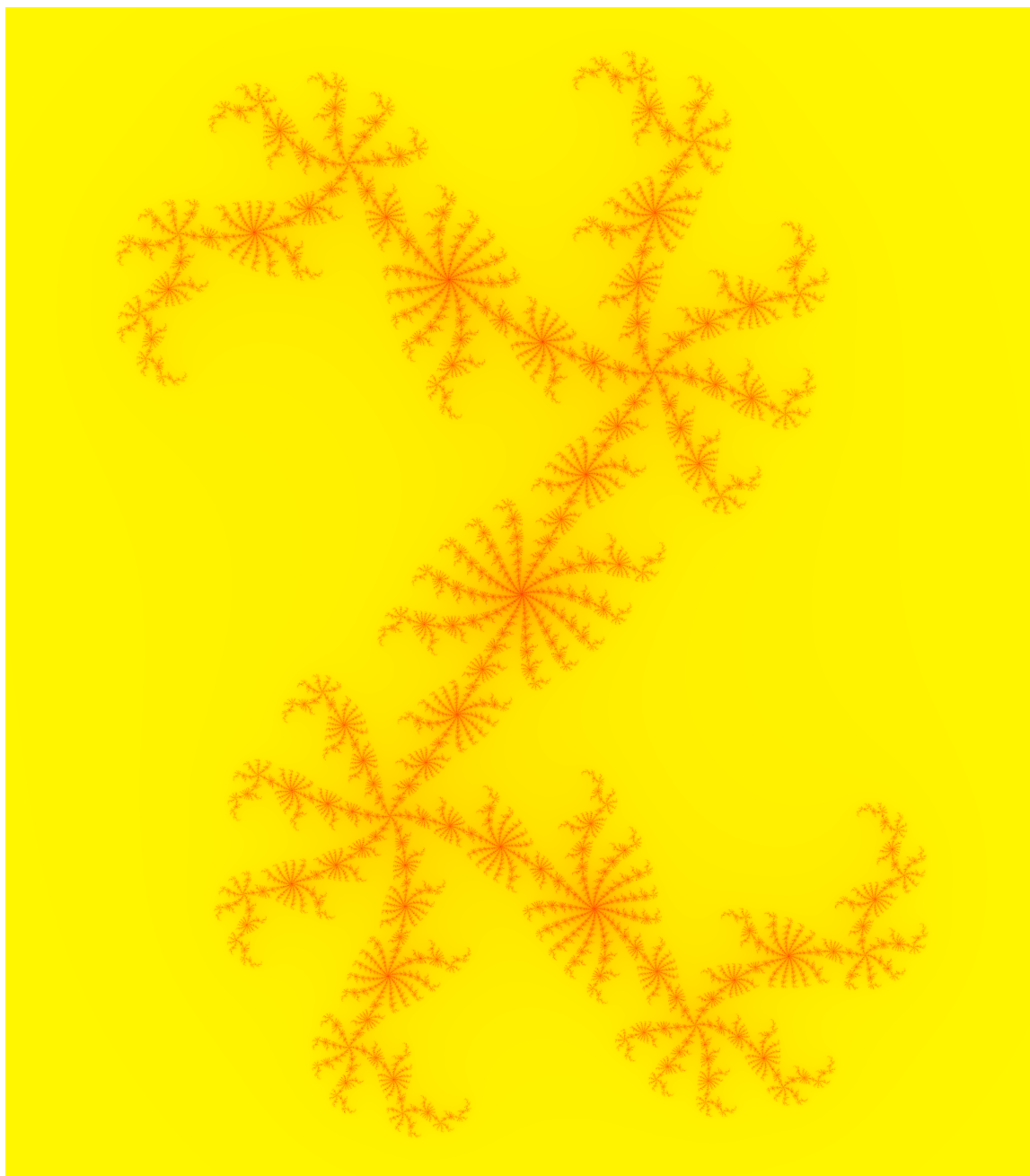


Figura 13: Conjunto de Julia para $z \mapsto z^2 + 0.424513 + 0.207530i$.

En la Figura 14 se muestra, por un lado, en la imagen izquierda un zoom del conjunto de Julia de la Figura 13, dibujado en la región $(-0.1, 0.1) \times (-0.1, 0.1)$, en el que se puede

apreciar con más detalle la estructura espiral logarítmica del mismo. Y en el otro dibujo se muestra un subrectángulo muy pequeño que reproduce parte de la figura total, lo que indica la propiedad de autosimilaridad que tiene esta estructura fractal.

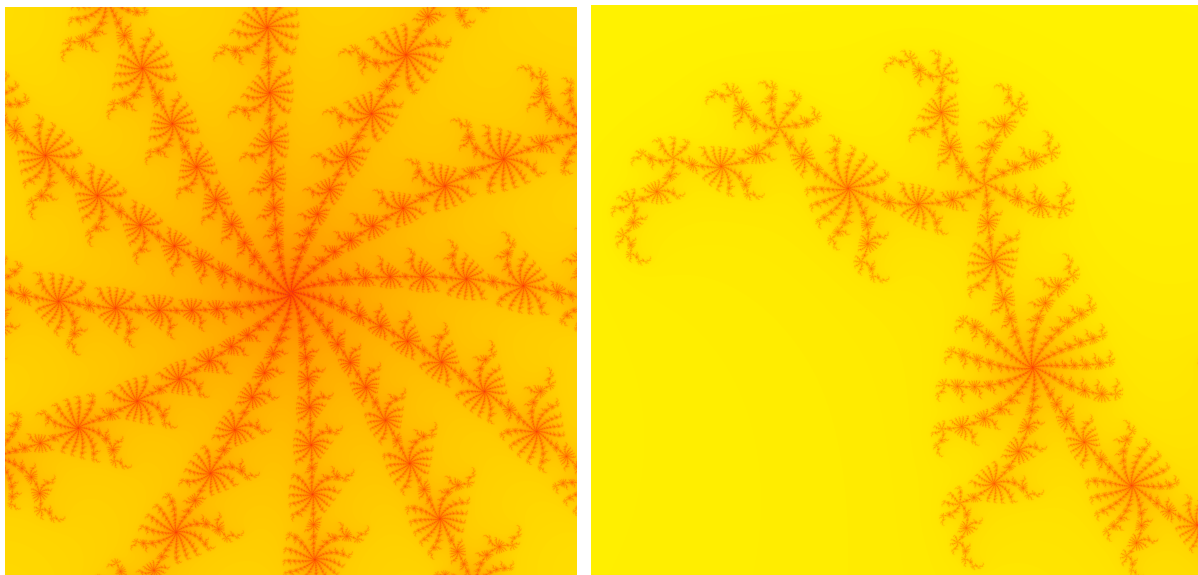


Figura 14: Detalles del conjunto de Julia para $z \mapsto z^2 + 0.424513 + 0.207530i$.

4.10. Conjunto de Mandelbrot

Benoît Mandelbrot (1924-2010) fue un matemático polaco considerado el responsable del actual interés por la geometría fractal debido a la introducción de los ordenadores en este campo. Mostró además cómo los fractales pueden aparecer en muchos ámbitos diferentes, no sólo de matemáticas, sino también en numerosos aspectos de la naturaleza, recogidos en su libro “The Fractal Geometry of Nature” publicado en 1982 [8]. A pesar de que el conjunto que lleva su nombre no fue descubierto por él, sí fue el primero en proporcionar una visualización computacional del mismo.



Figura 15:
Benoît Mandelbrot.

Una de las características de las funciones racionales mostradas en los ejemplos de conjuntos de Julia en las subsecciones anteriores (salvo los ejemplos correspondientes a las subsecciones 4.3 y 4.7) es que son funciones polinómicas f_c , $c \in \mathbb{C}$, de la forma:

$$f_c : z \mapsto z^2 + c$$

o, equivalentemente, utilizando pares homogéneos:

$$f_c : [z : t] \mapsto [z^2 + ct^2 : t^2].$$

Notemos que para cada $c \in \mathbb{C}$, la derivada esférica $f_c^\# : \hat{\mathbb{C}} \rightarrow \mathbb{R}$ viene dada por $f_c^\#(z) = 2|z| \frac{1+|z|^2}{1+|z^2+c|^2}$, o expresada como $f_c^\# : \mathbf{P}^1(\mathbb{C}) \rightarrow \mathbb{R}$ con polinomios bivariados:

$$[z : t] \mapsto \frac{2|z||t^3| + 2|z^3||t|}{|t|^4 + |z^2 + c| |t|^2}.$$

Los puntos fijos de f_c son: ∞ , $c_1 = \frac{-1+\sqrt{1-4c}}{2}$ y $c_2 = \frac{-1-\sqrt{1-4c}}{2}$. El punto ∞ es super-atractor ($f_c^\#(\infty) = 0$) y sabemos que c_1 y c_2 no pueden ser simultáneamente atractores, aplicando la Proposición 4.1.1.

Además, los puntos críticos de f_c son 0 e ∞ . Sabemos que la órbita de ∞ es (∞) y la órbita $(f_c^n(0))_{n \in \mathbb{N}}$ del punto crítico 0 es: $(0, c, c^2 + c, (c^2 + c)^2 + c, ((c^2 + c)^2 + c)^2 + c, \dots)$.

El análisis de esta sucesión según los valores de c es lo que origina la definición que damos a continuación.

Definición 4.10.1. Sea $\{f_c\}_{c \in \mathbb{C}}$ la familia de polinomios cuadráticos dada por $f_c(z) = z^2 + c$. Llamaremos conjunto de Mandelbrot de dicha familia al conjunto $\mathcal{M} = \{c \in \mathbb{C} \mid (f_c^n(0))_{n \in \mathbb{N}} \text{ es acotada}\}$.

Fueron los matemáticos A. Douady y J. H. Hubbard quienes probaron que el conjunto \mathcal{M} es conexo con complementario conexo, a pesar de que Mandelbrot había concluido experimentalmente que se trataba de un conjunto desconexo. En cuanto a la conexión local de \mathcal{M} , este último conjeturó que era un conjunto localmente conexo. Muchos son los matemáticos que han intentado demostrarlo, pero la conjetura sigue abierta. Para más detalles sobre el conjunto de Mandelbrot pueden verse [1] y [11].

Una de las características importantes de la sucesión $(f_c^n(0))_{n \in \mathbb{N}}$, pues ayuda a la hora de implementar los algoritmos, es que si un punto de la sucesión verifica que su distancia a 0 es mayor que 2 (esto es, $|f_c^n(0)| > 2$ para algún $n \in \mathbb{N}$), entonces el punto c no pertenece al conjunto de Mandelbrot.

En la Figura 16 se muestra una imagen del conjunto de Mandelbrot, dibujado en la región $(-2.0, 0.8) \times (-1.2, 1.2)$, obtenido mediante los algoritmos implementados en Julia.

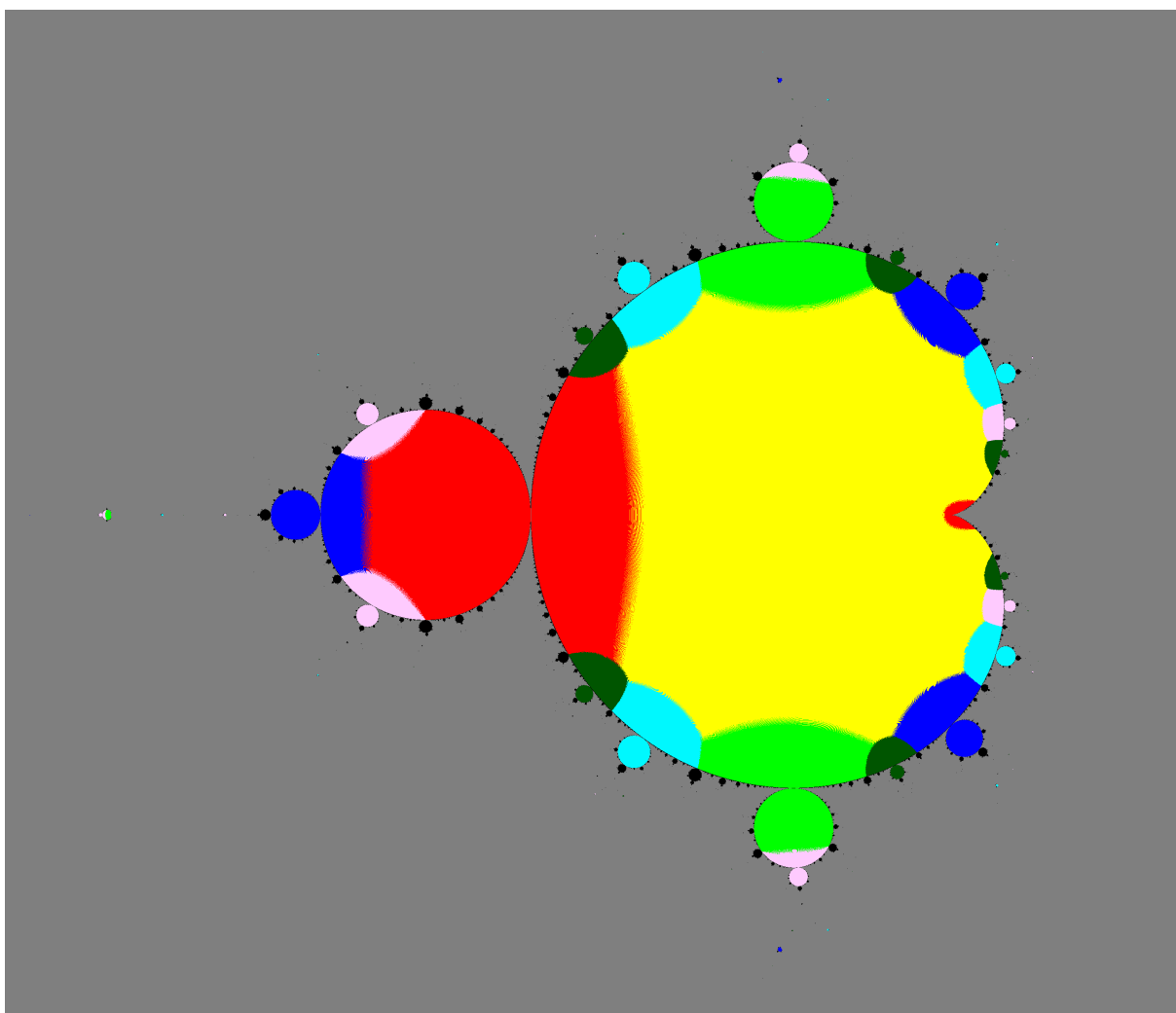


Figura 16: Conjunto de Mandelbrot para las funciones $f_c(z) = z^2 + c$.

En dicha imagen, el conjunto \mathcal{M} aparece como una reunión de lóbulos, donde, curiosamente, el único que posee forma cardioidal es el lóbulo principal. Además, cabe destacar que, teniendo en cuenta algunas propiedades conocidas del conjunto de Mandelbrot, cada lóbulo debería poseer un único color. Por ejemplo, un lóbulo de color amarillo para aquellos puntos $c \in \mathbb{C}$ cuya sucesión asociada $(f_c^n(0))_{n \in \mathbb{N}}$ converge a algún 1-ciclo, el color rojo para los c cuya sucesión correspondiente converge a algún 2-ciclo, el verde para indicar los asociados a 3-ciclos, etc.

Sin embargo, dado que estamos trabajando en un ámbito computacional con ciertas tolerancias prefijadas y un número máximo de iteraciones, esto no se ve reflejado en la gráfica mostrada. Por ejemplo, la parte roja del cardioide principal debería también ser amarilla. El color rojo con el que aparecen esos puntos es probable que se deba a una convergencia de forma 2-cíclica muy lenta al punto fijo correspondiente y es por ello que el algoritmo que hemos implementado en Julia interpreta que convergen a un 2-ciclo, a pesar de que la sucesión asociada a cada uno de dichos puntos converge a algún punto fijo, lo mismo que ocurre con el resto de los puntos de dicho lóbulo.

Conclusiones

A principios del primer trimestre, tras la aceptación por parte de mis tutores de la petición que les hice de realizar un Trabajo Fin de Grado relacionado con Topología, el tema propuesto por ellos fue la representación de Conjuntos de Julia mediante el lenguaje de programación Julia. Esta sugerencia me llamó la atención, no sólo por ser éste un asunto relacionado con Topología, sino porque ellos me dijeron que con él podría tener un primer contacto con la investigación. Además, me pareció muy acertada la idea de poder combinar una parte teórica y otra práctica en mi TFG.

En la parte teórica, he podido utilizar algunos de los conocimientos adquiridos durante el grado relacionados con topología, análisis complejo, geometría, programación y análisis numérico, entre otros, pero también ampliarlos gracias al estudio realizado en el trabajo. Además, ha sido muy interesante ver cómo unos modelos teóricos de tipo topológico y geométrico pueden ser usados para dar un marco matemático adecuado en la elaboración de algoritmos y poder solventar con esas estructuras los problemas que se presentan a la hora de realizar cálculos computacionales con funciones racionales.

En la parte práctica, he podido aprender el lenguaje de programación de alto nivel Julia y estudiar e implementar algoritmos en él, algunos de ellos utilizando programación en paralelo. Ha sido muy motivador el poder hacer uso de éstos y del entorno informático Julia para realizar y analizar distintas representaciones gráficas de Conjuntos de Julia.

Haciendo una reflexión, creo que la elaboración de este trabajo ha sido una buena forma de acabar el grado, ya que me ha ayudado a afianzar conceptos y también a adquirir nuevas habilidades al tener que buscar información, leer y entender artículos, y redactar de forma adecuada. Además, no sólo he aprendido un nuevo lenguaje de programación, sino técnicas informáticas como la programación en paralelo que tienen importantes aplicaciones en numerosos campos científicos y sociales, y, además, serán de gran utilidad en un futuro próximo.

Referencias

- [1] A. F. Beardon, *Iteration of Rational Functions*, Complex Analytic Dynamical Systems, Springer-Verlag, New York, 1991.
- [2] G. D. Birkhoff, *Dynamical System*, American Mathematical Society, Colloquium Publications, vol. 9, 1927.
- [3] M. P. do Carmo, *Geometría diferencial de curvas y superficies*, Alianza Editorial, 1990.
- [4] J. I. Extremiana, J. M. Gutiérrez, L. J. Hernández y M. T. Rivas, *A graphic method for detecting multiple roots based on the nullity tolerance*, preprint, 2018.
- [5] P. J. L. Fatou, *Sur les équations fonctionnelles*, Bulletin de la Société Mathématique de France, tome 47, 1919, pp. 161-271.
- [6] L. J. Hernández, *Global Visualization of Basins of Univariate Rational Functions with Julia*, User manual of the package GVBURF06.jl, <https://github.com/luisjavierhernandez>.
- [7] G. M. Julia, *Mémoire sur l'itération des fonctions rationnelles*, Journal de Mathématiques Pures et Appliquées, 8^e série, tome 1, 1918, pp. 47-246.
- [8] B. Mandelbrot, *The Fractal Geometry of Nature*, W. H. Freeman and Company, 1982.
- [9] W. S. Massey, *Algebraic Topology: An Introduction*, Springer, 1991.
- [10] J. W. Milnor, *Dynamics in One Complex Variable: Introductory Lectures*, Stony Brook IMS, 1990, [arXiv:math/9201272](https://arxiv.org/abs/math/9201272).
- [11] J. W. Milnor, *Local connectivity of Julia sets: Expository Lectures*, Stony Brook IMS, 1992, [arXiv:math/9207220](https://arxiv.org/abs/math/9207220).
- [12] J. W. Milnor, *Remarks on Quadratic Rational Maps*, Stony Brook IMS, 1992, [arXiv:math/9209221](https://arxiv.org/abs/math/9209221).
- [13] J. H. Poincaré, *Mémoire sur les courbes définies par une équation différentielle*, Journal de Mathématiques Pures et Appliquées, 3^e série, tome 7, 1886, pp. 375-422.

- [14] W. A. Sutherland, *Introduction to Metric and Topological Spaces*, Oxford University Press, 2010.
- [15] C. Thomassen, *The Jordan-Schönflies Theorem and the Classification of Surfaces*, The American Mathematical Monthly, vol. 99, 1992, pp. 116-131.
- [16] O. Veblen, *Theory on Plane Curves in Non-Metrical Analysis Situs*, Transactions of the American Mathematical Society, vol. 6, 1905, pp. 83-98, [doi:10.2307/1986378](https://doi.org/10.2307/1986378).
- [17] S. Willard, *General Topology*, Addison-Wesley, 1970.
- [18] *The Julia Language*, <https://julialang.org>.
- [19] *JuliaBox, Run Julia in your Browser*, <https://juliabox.com>.
- [20] *Julia Computing*, <https://juliacomputing.com>.